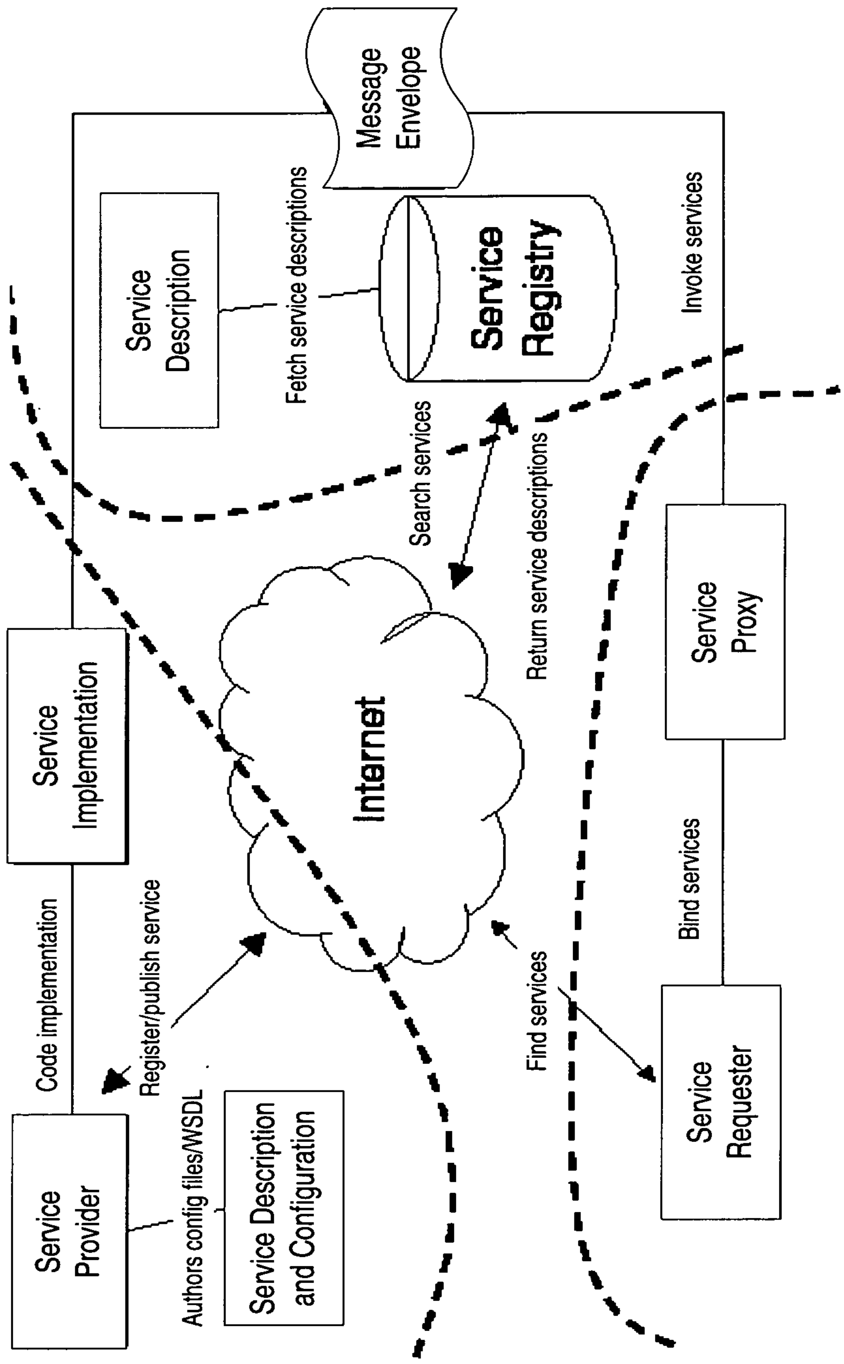

Message Envelope
Service Description
Fetch service descriptions
Service Registry
Invoke services
Search services
Return service descriptions
Service Implementation
Internet
Service Proxy
Code implementation
Register/publish service
Bind services
Find services
Service Provider
Authors config files/WSDL
Service Description and Configuration
Service Requester

FIG. 1

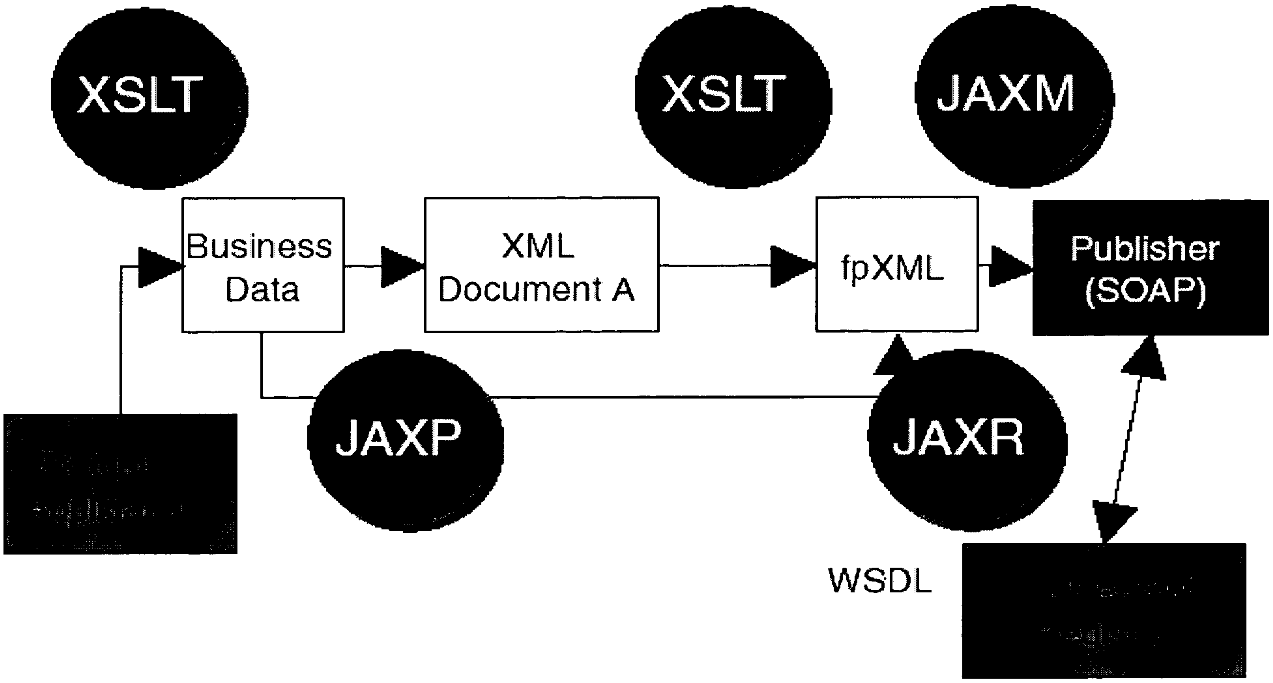
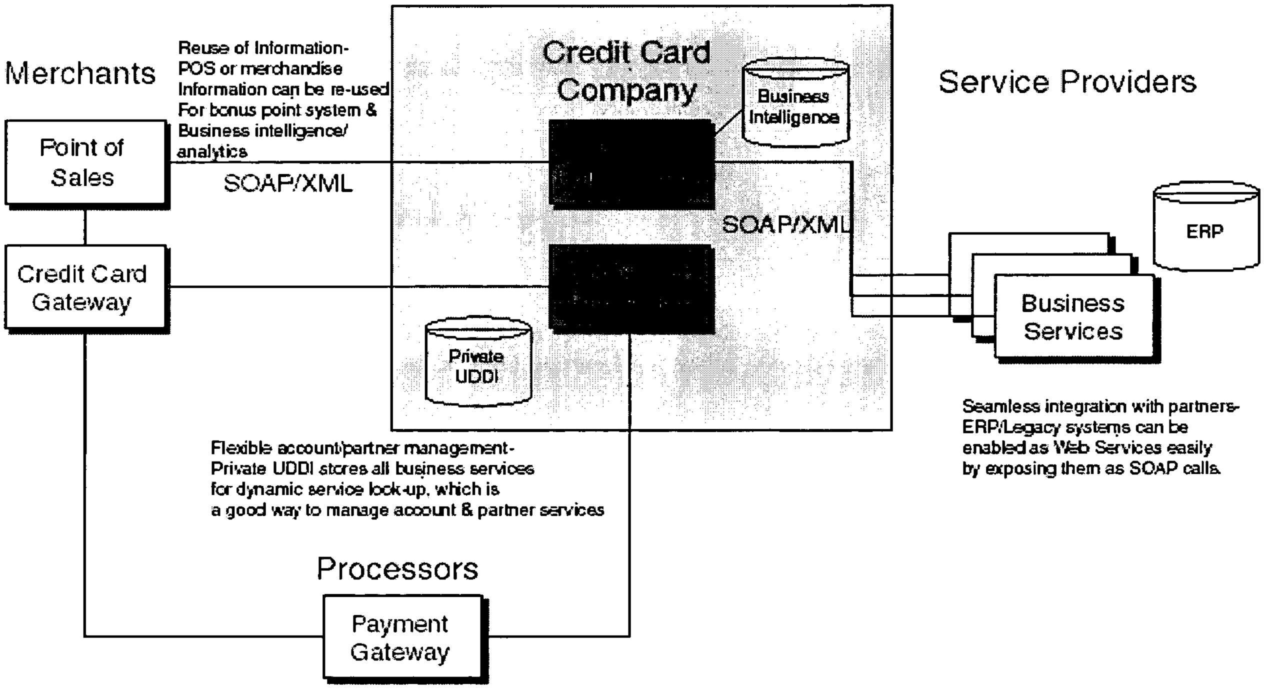


FIG. 2



Reuse of Information-
POS or merchandise
Information can be re-used
For bonus point system &
Business intelligence/
analytics

Seamless integration with partners-
ERP/Legacy systems can be
enabled as Web Services easily
by exposing them as SOAP calls.

Flexible account/partner management-
Private UDDI stores all business services
for dynamic service look-up, which is
a good way to manage account & partner services

FIG. 3

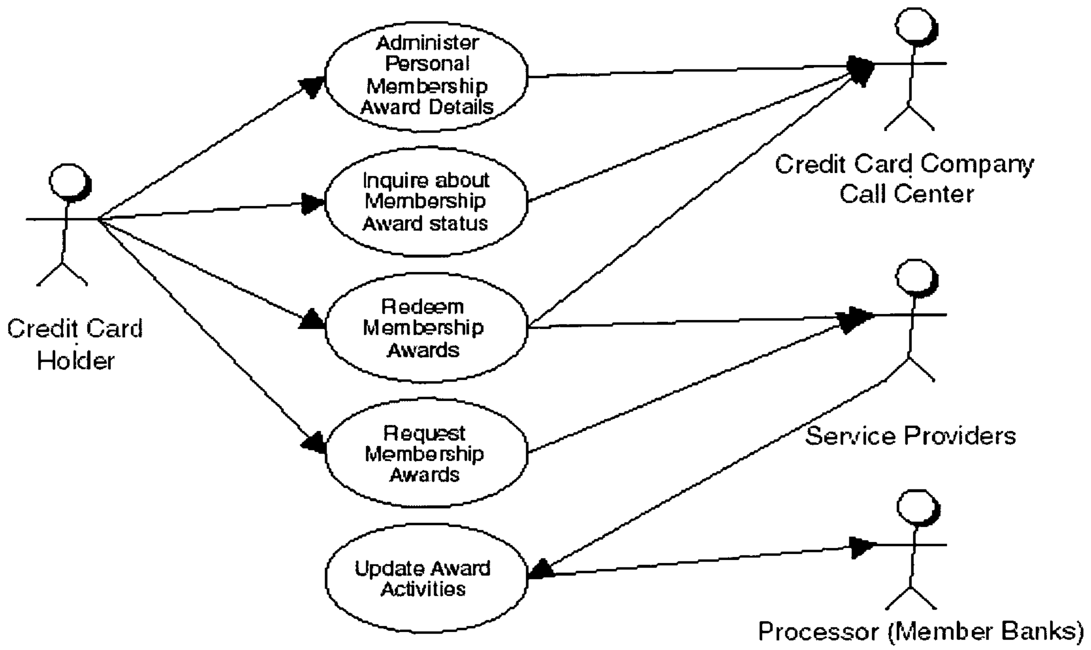

Administer Personal Membership Award Details
Inquire about Membership Award status
Redeem Membership Awards
Request Membership Awards
Update Award Activities
Credit Card Holder
Credit Card Company Call Center
Service Providers
Processor (Member Banks)

FIG. 4

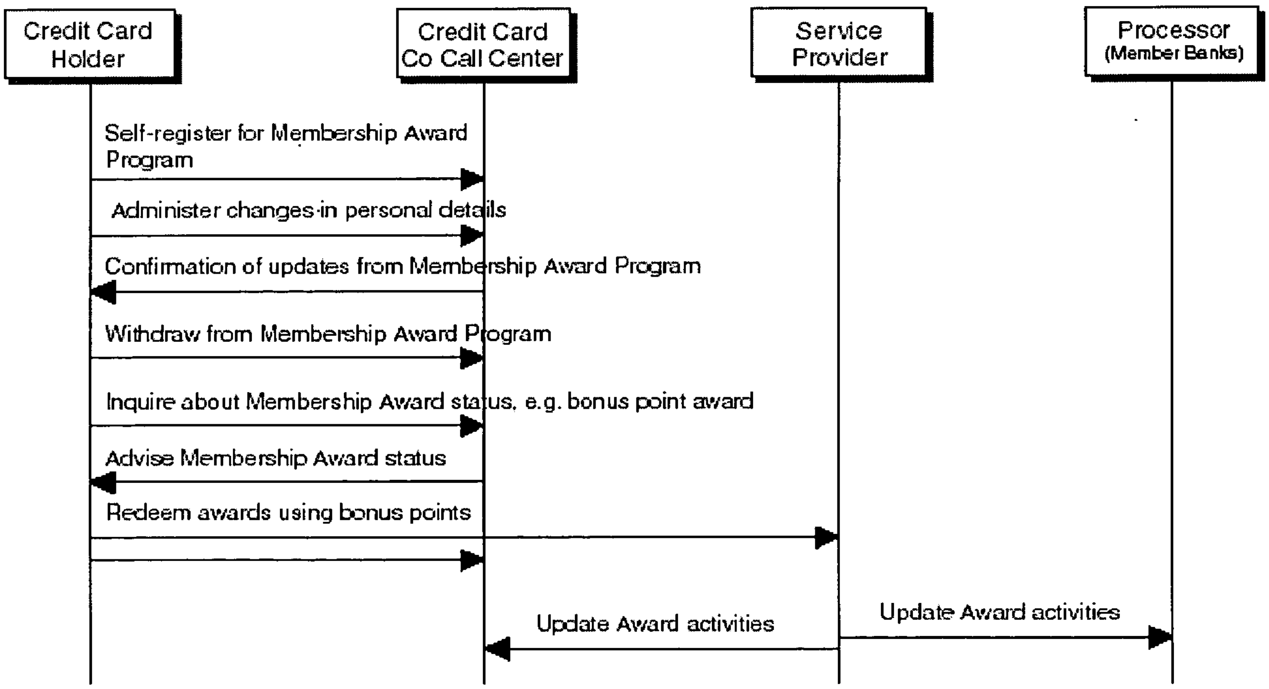

Credit Card Holder
Credit Card Co Call Center
Service Provider
Processor (Member Banks)
Self-register for Membership Award Program
Administer changes in personal details
Confirmation of updates from Membership Award Program
Withdraw from Membership Award Program
Inquire about Membership Award status, e.g. bonus point award
Advise Membership Award status
Redeem awards using bonus points
Update Award activities
Update Award activities

FIG. 5

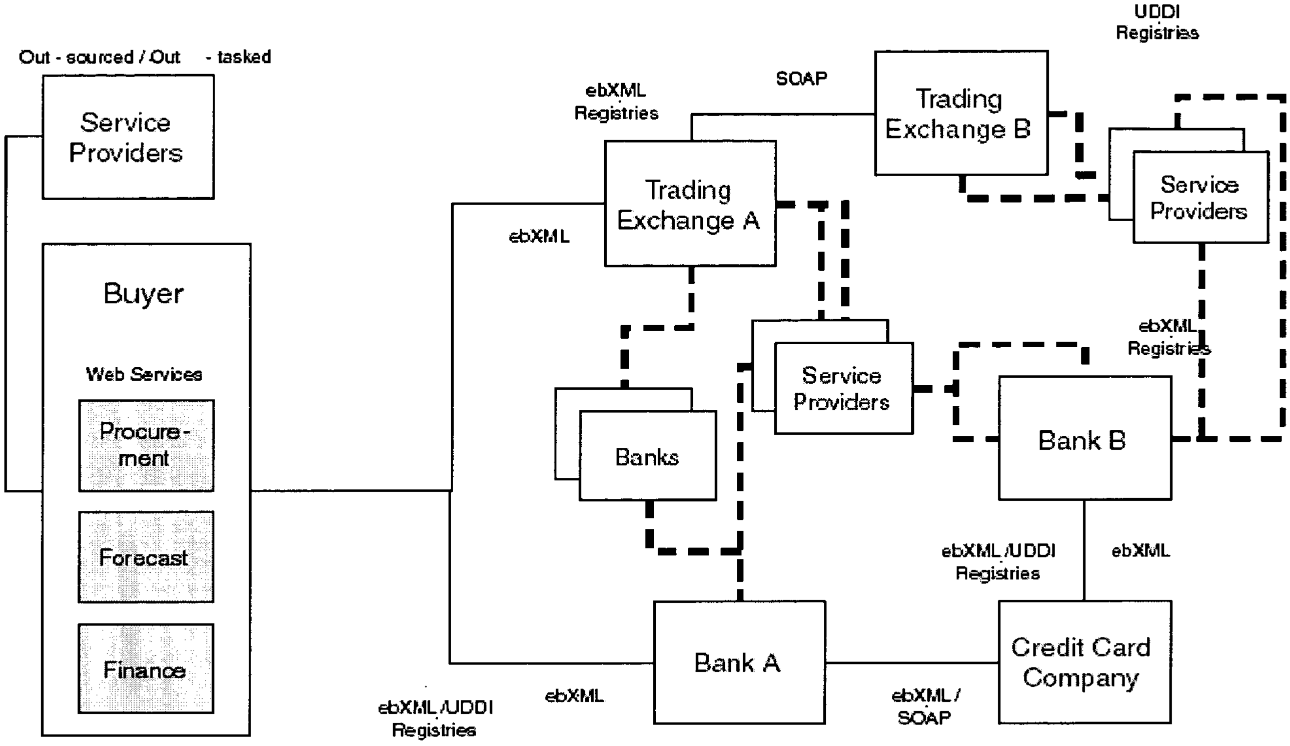

Out - sourced / Out - tasked
Service Providers
Buyer
Web Services
Procure - ment
Forecast
Finance
ebXML /UDDI Registries
ebXML
ebXML
ebXML Registries
Trading Exchange A
SOAP
Trading Exchange B
UDDI Registries
Service Providers
Banks
Service Providers
ebXML Registries
Bank B
ebXML /UDDI Registries
ebXML
Bank A
ebXML / SOAP
Credit Card Company

FIG. 6

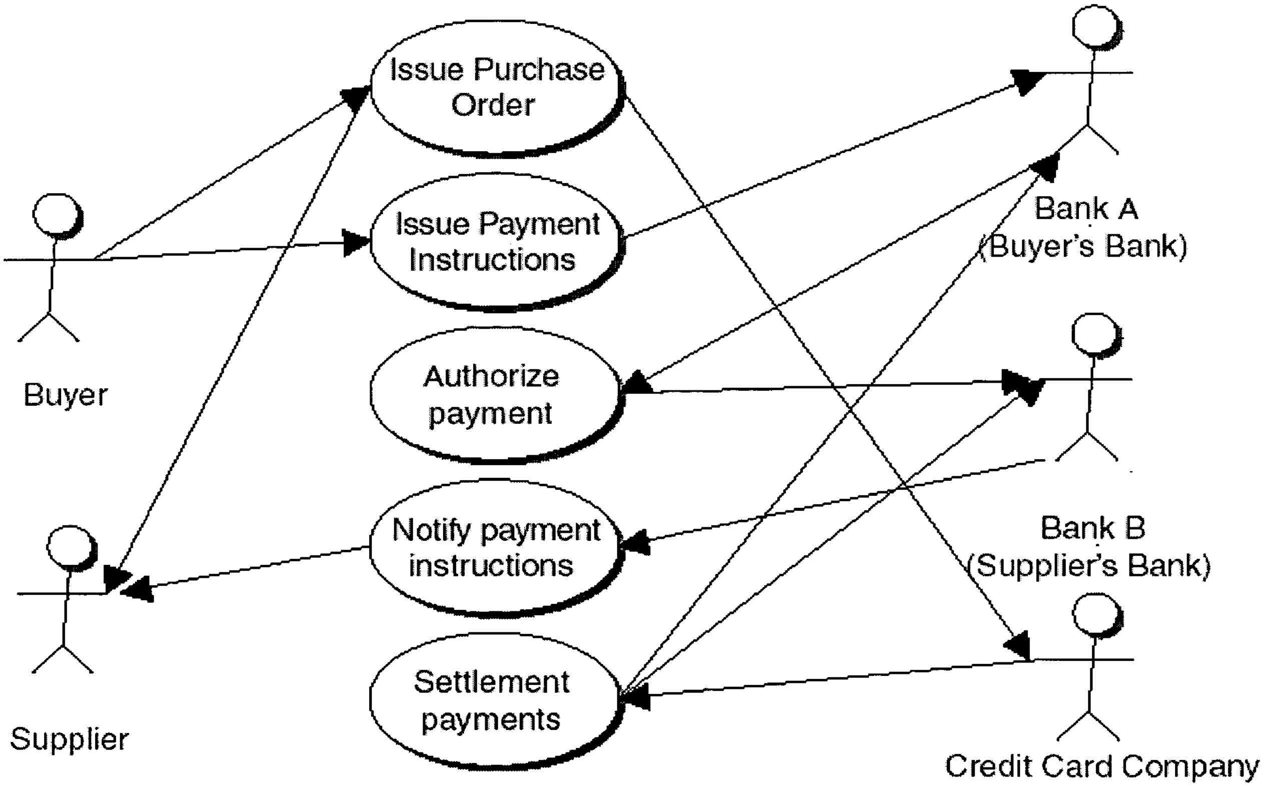

Issue Purchase Order
Issue Payment Instructions
Authorize payment
Notify payment instructions
Settlement payments
Buyer
Supplier
Bank A (Buyer's Bank)
Bank B (Supplier's Bank)
Credit Card Company

FIG. 7

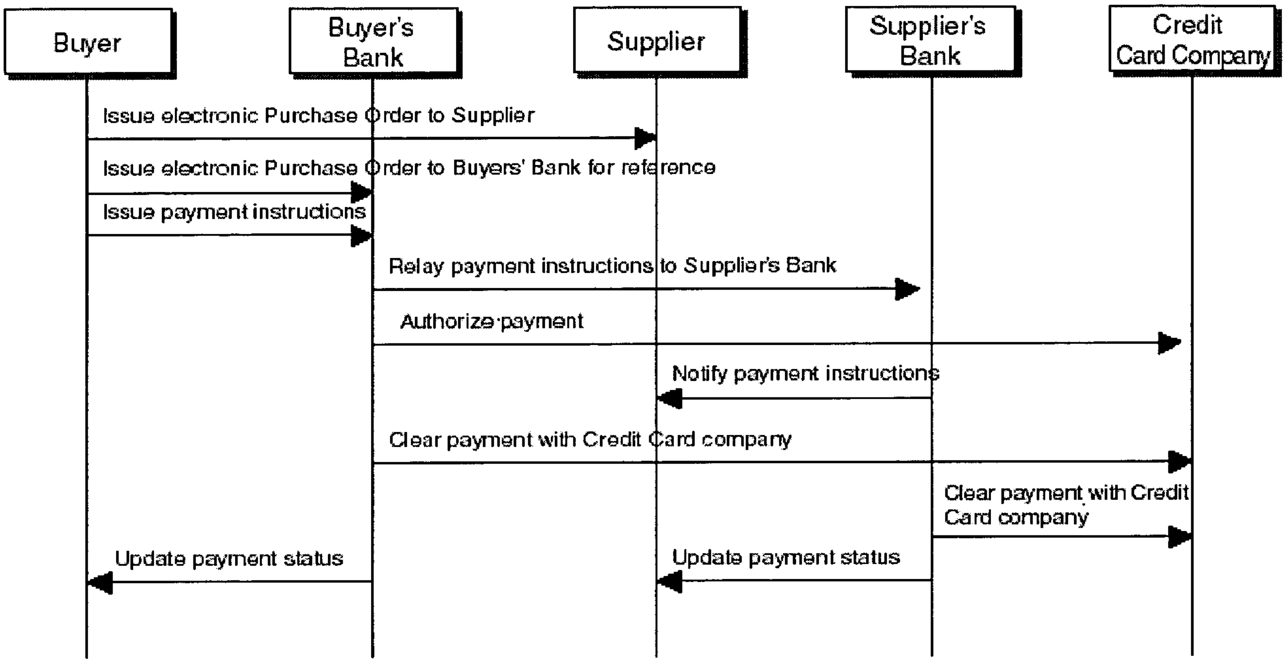

Buyer
Buyer's Bank
Supplier
Supplier's Bank
Credit Card Company
Issue electronic Purchase Order to Supplier
Issue electronic Purchase Order to Buyers' Bank for reference
Issue payment instructions
Relay payment instructions to Supplier's Bank
Authorize payment
Notify payment instructions
Clear payment with Credit Card company
Clear payment with Credit Card company
Update payment status
Update payment status

FIG. 8

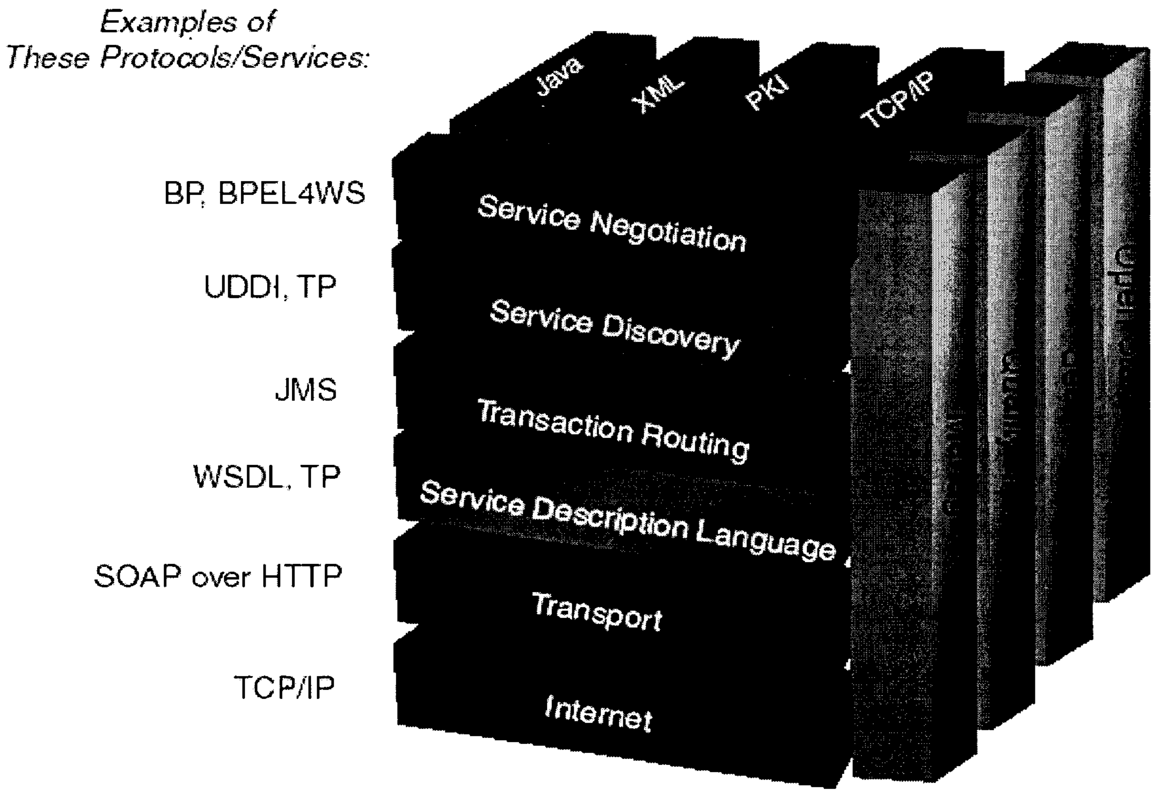

Examples of These Protocols/Services:
BP, BPEL4WS
UDDI, TP
JMS
WSDL, TP
SOAP over HTTP
TCP/IP
Java
XML
PKI
TCP/IP
Service Negotiation
Service Discovery
Transaction Routing
Service Description Language
Transport
Internet

FIG. 9

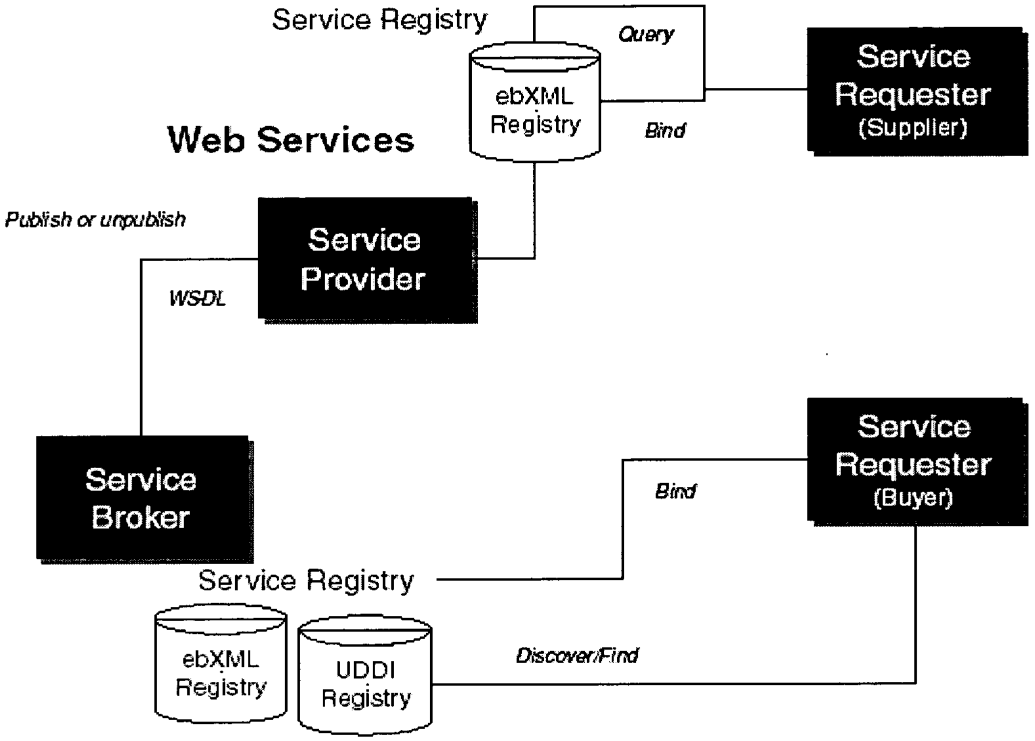

Service Registry
Query
ebXML Registry
Bind
Service Requester (Supplier)
Web Services
Publish or unpublish
Service Provider
WSDL
Service Broker
Service Requester (Buyer)
Bind
Service Registry
Discover/Find
ebXML Registry
UDDI Registry

FIG. 10

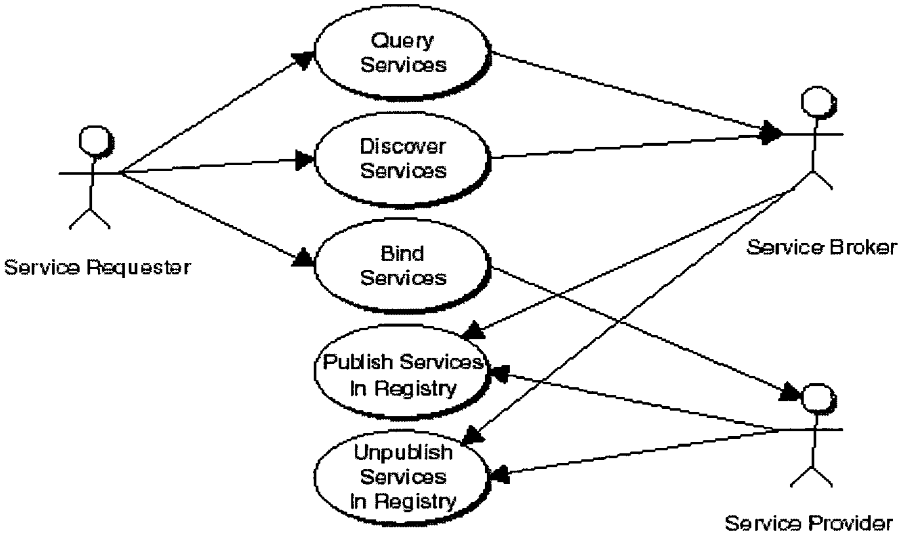

Query Services
Discover Services
Bind Services
Publish Services In Registry
Unpublish Services In Registry
Service Requester
Service Broker
Service Provider

FIG. 11

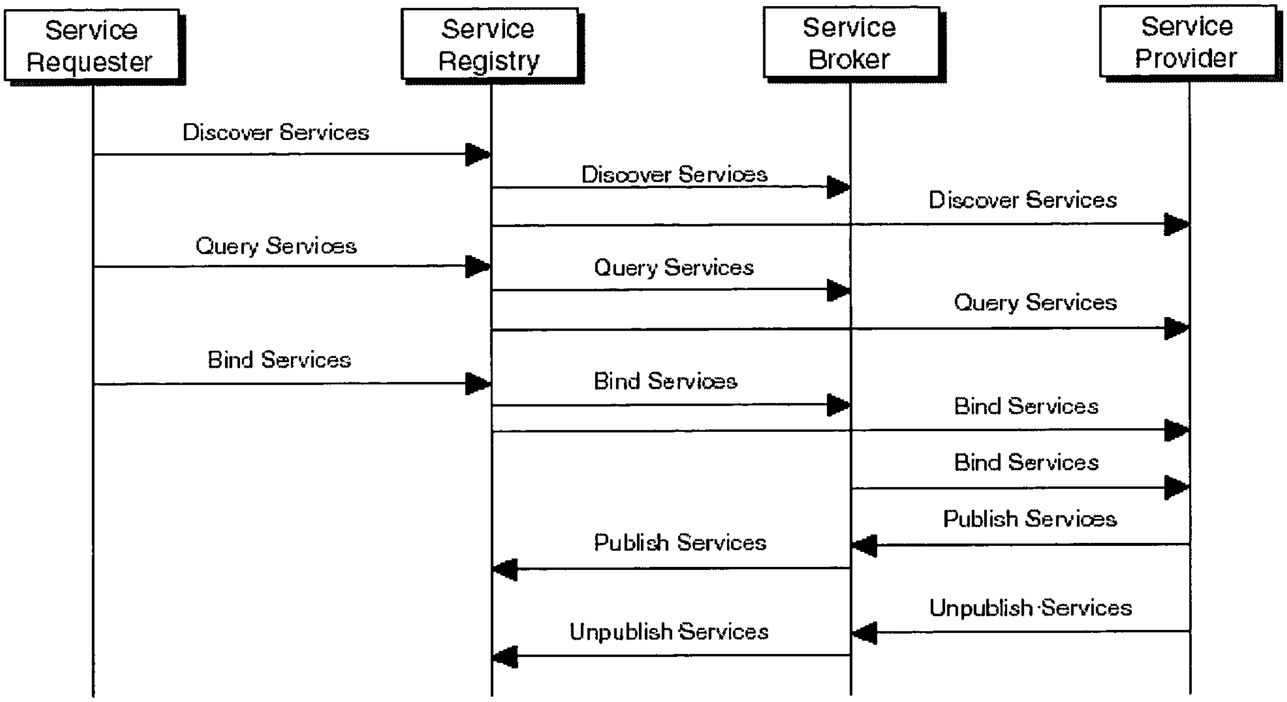

Service Requester
Service Registry
Service Broker
Service Provider
Discover Services
Discover Services
Discover Services
Query Services
Query Services
Query Services
Bind Services
Bind Services
Bind Services
Bind Services
Publish Services
Publish Services
Unpublish Services
Unpublish Services

FIG. 12

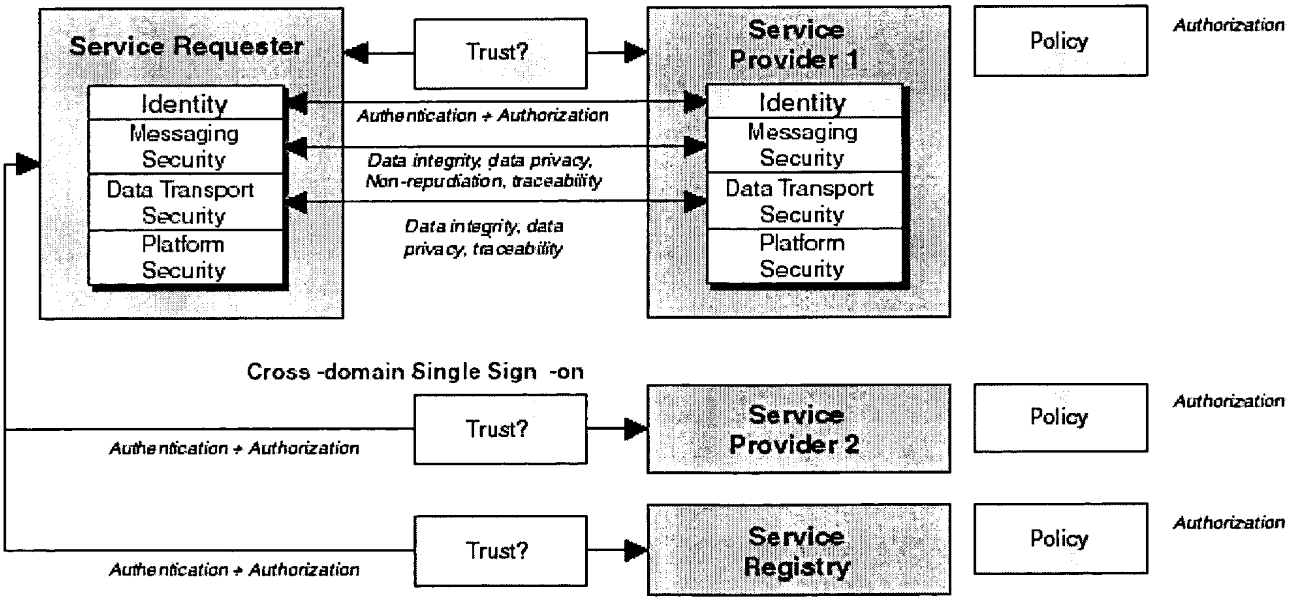

Service Requester
Identity
Messaging Security
Data Transport Security
Platform Security
Trust?
Service Provider 1
Identity
Messaging Security
Data Transport Security
Platform Security
Policy
Authorization
Authentication + Authorization
Data integrity, data privacy, Non-repudiation, traceability
Data integrity, data privacy, traceability
Cross -domain Single Sign -on
Authentication + Authorization
Trust?
Service Provider 2
Policy
Authorization
Authentication + Authorization
Trust?
Service Registry
Policy
Authorization

FIG. 13

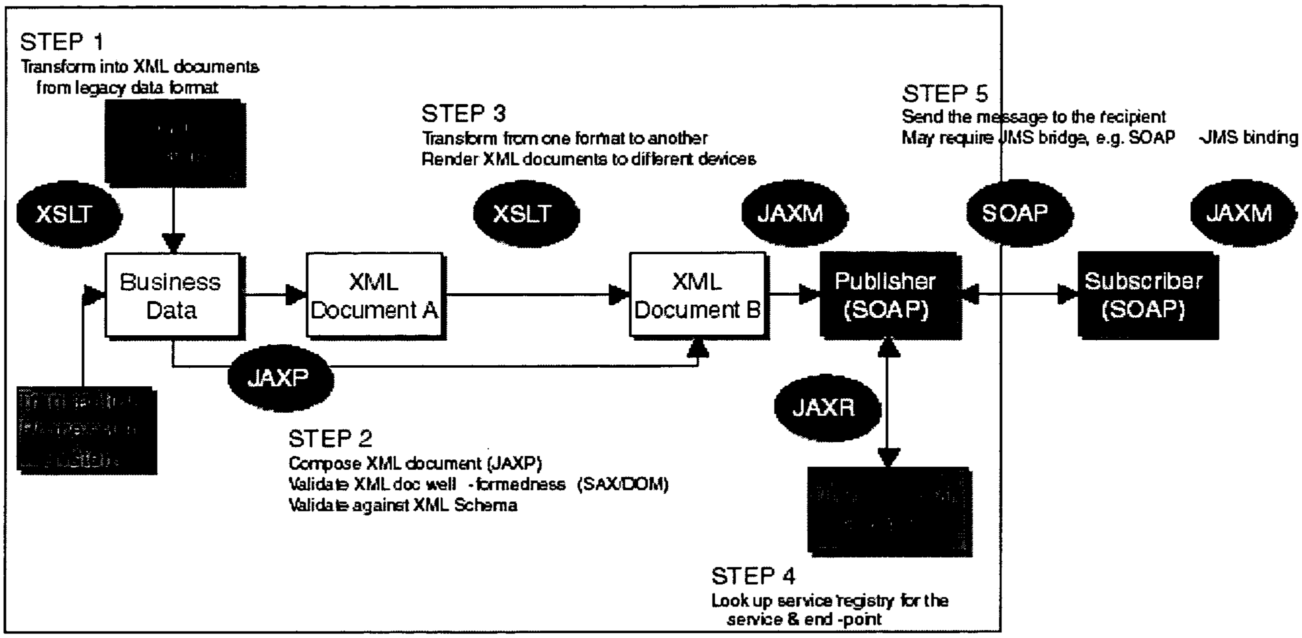

STEP 1
Transform into XML documents from legacy data format
XSLT
Business Data
XML Document A
STEP 3
Transform from one format to another
Render XML documents to different devices
XSLT
XML Document B
JAXM
Publisher (SOAP)
STEP 5
Send the message to the recipient
May require JMS bridge, e.g. SOAP -JMS binding
SOAP
JAXM
Subscriber (SOAP)
JAXP
STEP 2
Compose XML document (JAXP)
Validate XML doc well -formedness (SAX/DOM)
Validate against XML Schema
JAXR
STEP 4
Look up service registry for the service & end -point

FIG. 14

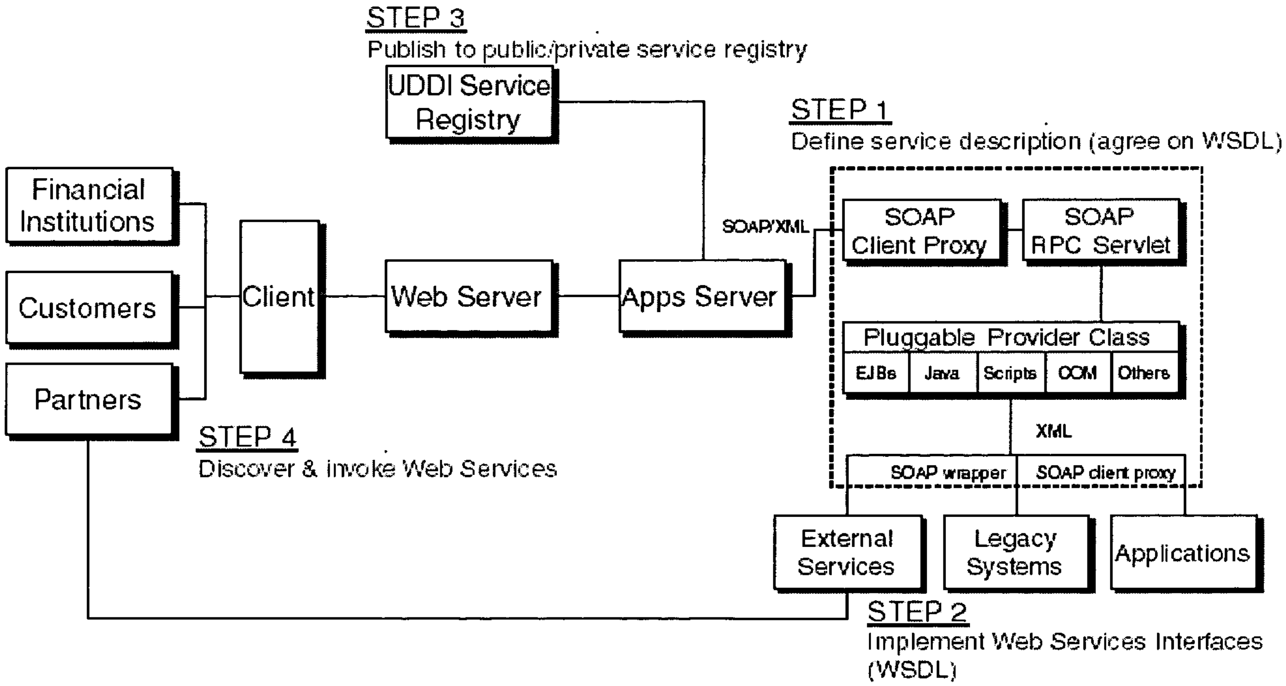

STEP 3
Publish to public/private service registry
UDDI Service Registry
STEP 1
Define service description (agree on WSDL)
Financial Institutions
Customers
Partners
Client
Web Server
Apps Server
SOAP/XML
SOAP Client Proxy
SOAP RPC Servlet
Pluggable Provider Class
EJBs
Java
Scripts
COM
Others
XML
SOAP wrapper
SOAP client proxy
STEP 4
Discover & invoke Web Services
External Services
Legacy Systems
Applications
STEP 2
Implement Web Services Interfaces (WSDL)

FIG. 15

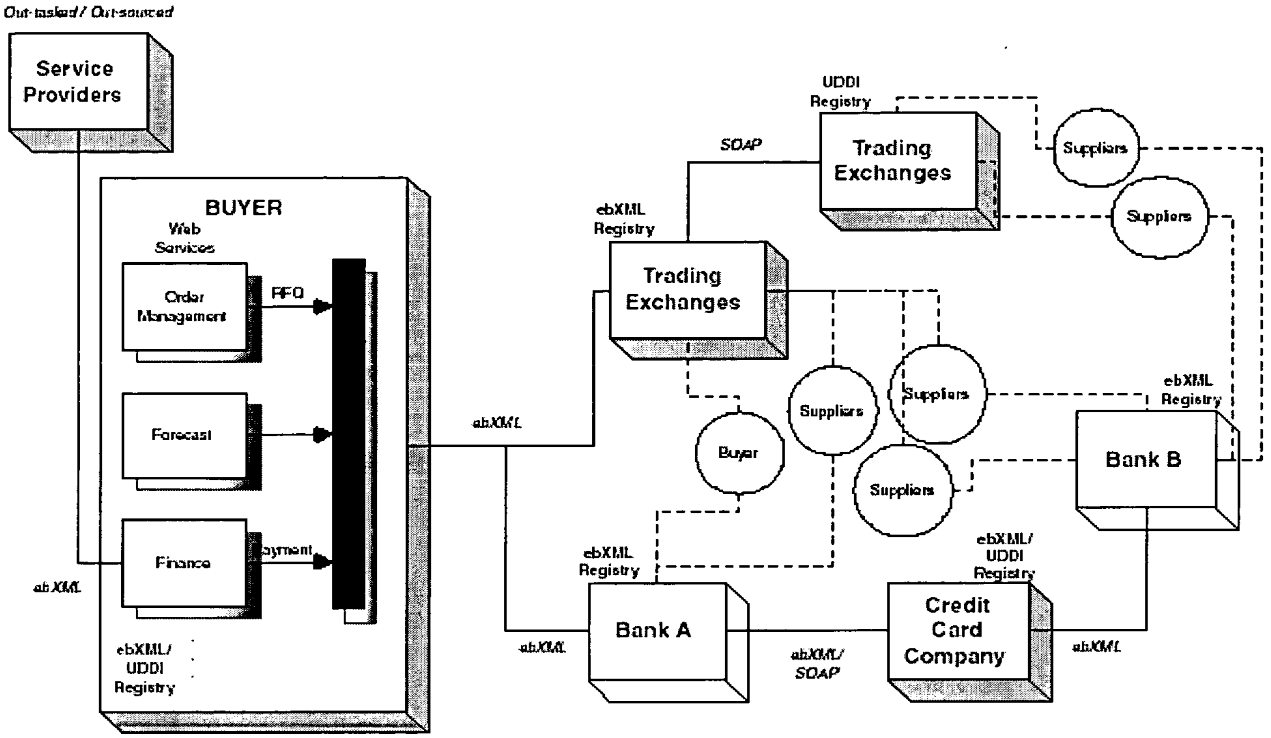

Out-tasked / Out-sourced
Service Providers
BUYER
Web Services
Order Management
RFQ
Forecast
Finance
Payment
ebXML
ebXML/ UDDI Registry
ebXML
SOAP
UDDI Registry
Trading Exchanges
ebXML Registry
Trading Exchanges
Suppliers
Suppliers
Suppliers
Suppliers
Suppliers
Buyer
ebXML Registry
Bank B
ebXML Registry
Bank A
ebXML
ebXML/ UDDI Registry
Credit Card Company
ebXML/ SOAP
ebXML

FIG. 16

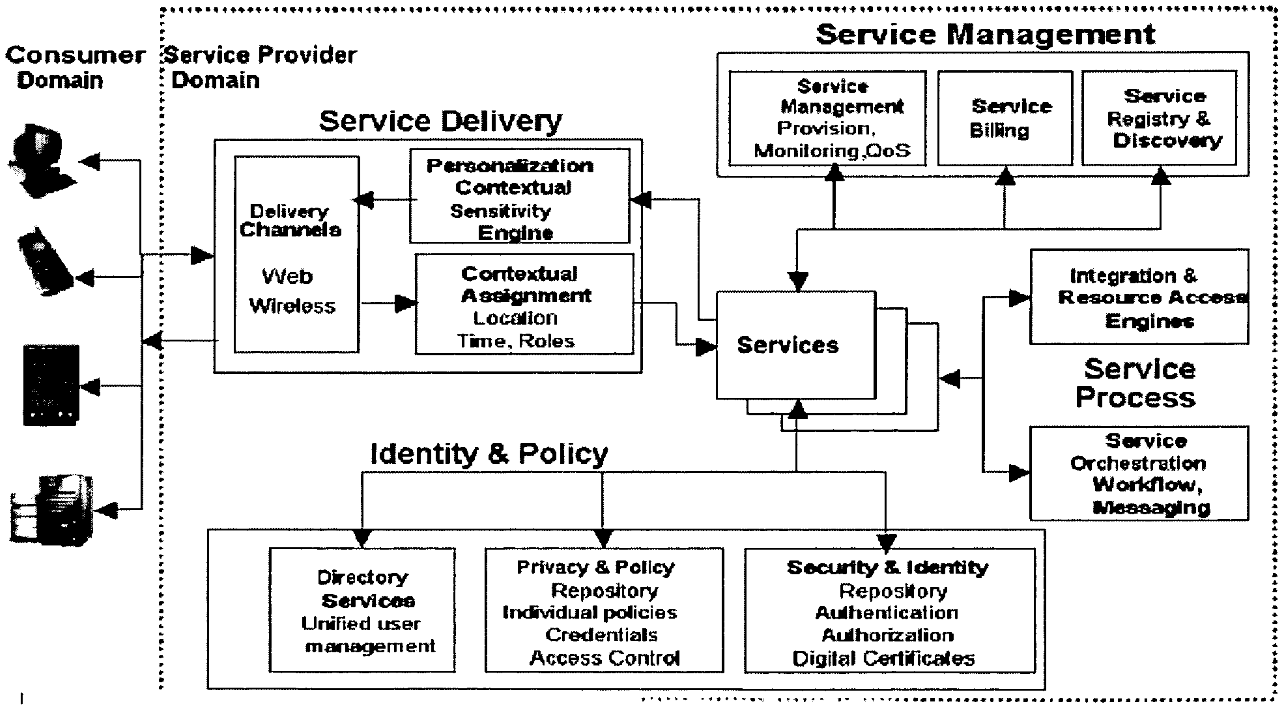

Consumer Domain
Service Provider Domain
Service Management
Service Management Provision, Monitoring,QoS
Service Billing
Service Registry & Discovery
Service Delivery
Delivery Channels
Web
Wireless
Personalization Contextual Sensitivity Engine
Contextual Assignment Location Time, Roles
Services
Integration & Resource Access Engines
Service Process
Service Orchestration Workflow, Messaging
Identity & Policy
Directory Services Unified user management
Privacy & Policy Repository Individual policies Credentials Access Control
Security & Identity Repository Authentication Authorization Digital Certificates

FIG. 17

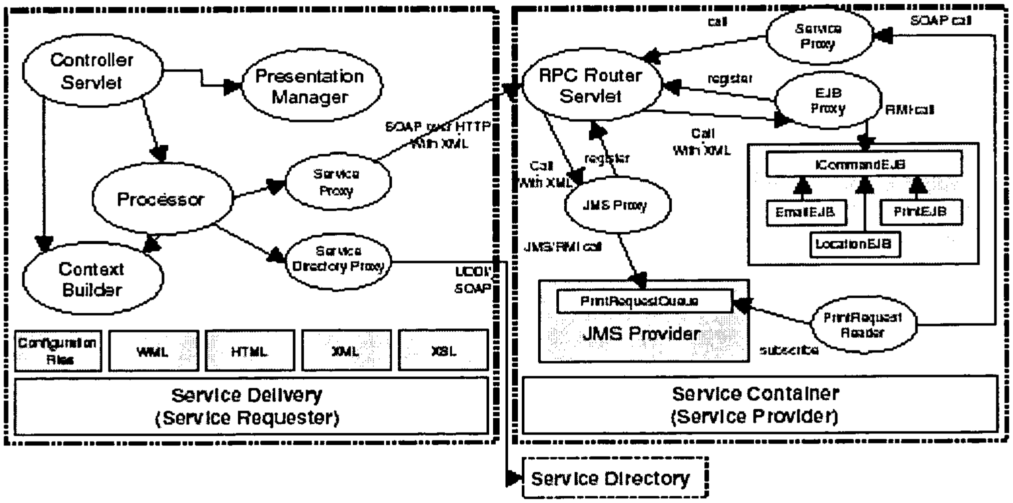
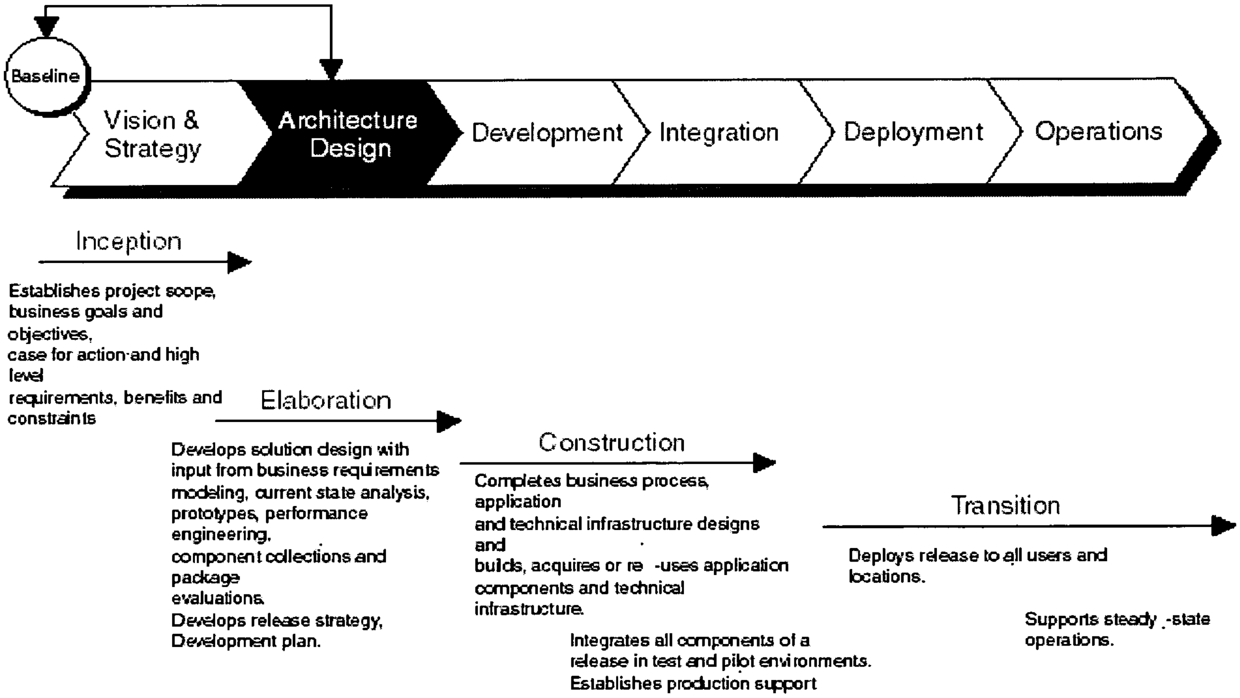


FIG. 18



Establishes project scope,
business goals and
objectives,
case for action and high
level
requirements, benefits and
constraints

Develops solution design with
input from business requirements
modeling, current state analysis,
prototypes, performance
engineering,
component collections and
package
evaluations.
Develops release strategy,
Development plan.

Completes business process,
application
and technical infrastructure designs
and
builds, acquires or re -uses application
components and technical
infrastructure.

Integrates all components of a
release in test and pilot environments.
Establishes production support

Deploys release to all users and
locations.

Supports steady -state
operations.

FIG. 19

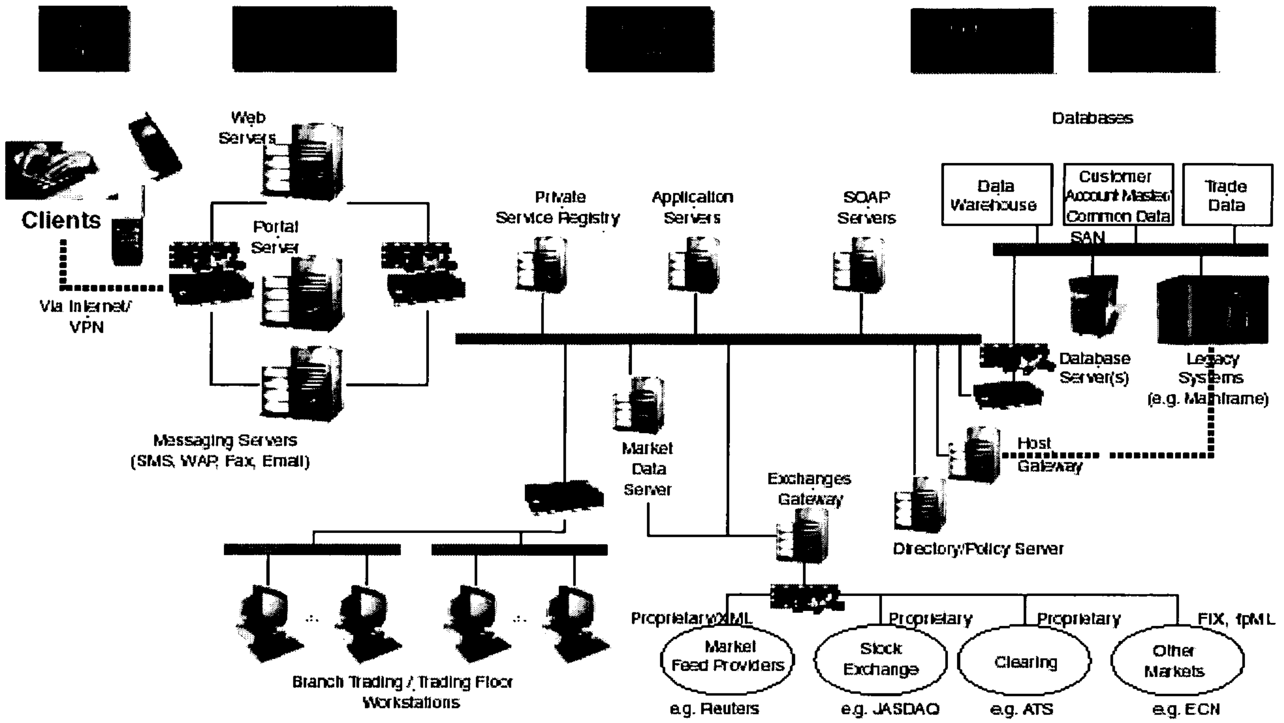

Databases
Web Servers
Clients
Portal Server
Private Service Registry
Application Servers
SOAP Servers
Data Warehouse
Customer Account Master/ Common Data
Trade Data
SAN
Via Internet/ VPN
Database Server(s)
Legacy Systems (e.g. Mainframe)
Messaging Servers (SMS, WAP, Fax, Email)
Market Data Server
Host Gateway
Exchanges Gateway
Directory/Policy Server
Proprietary/XML
Proprietary
Proprietary
FIX, fpML
Market Feed Providers
Stock Exchange
Clearing
Other Markets
Branch Trading / Trading Floor Workstations
e.g. Reuters
e.g. JASDAQ
e.g. ATS
e.g. ECN

FIG. 20

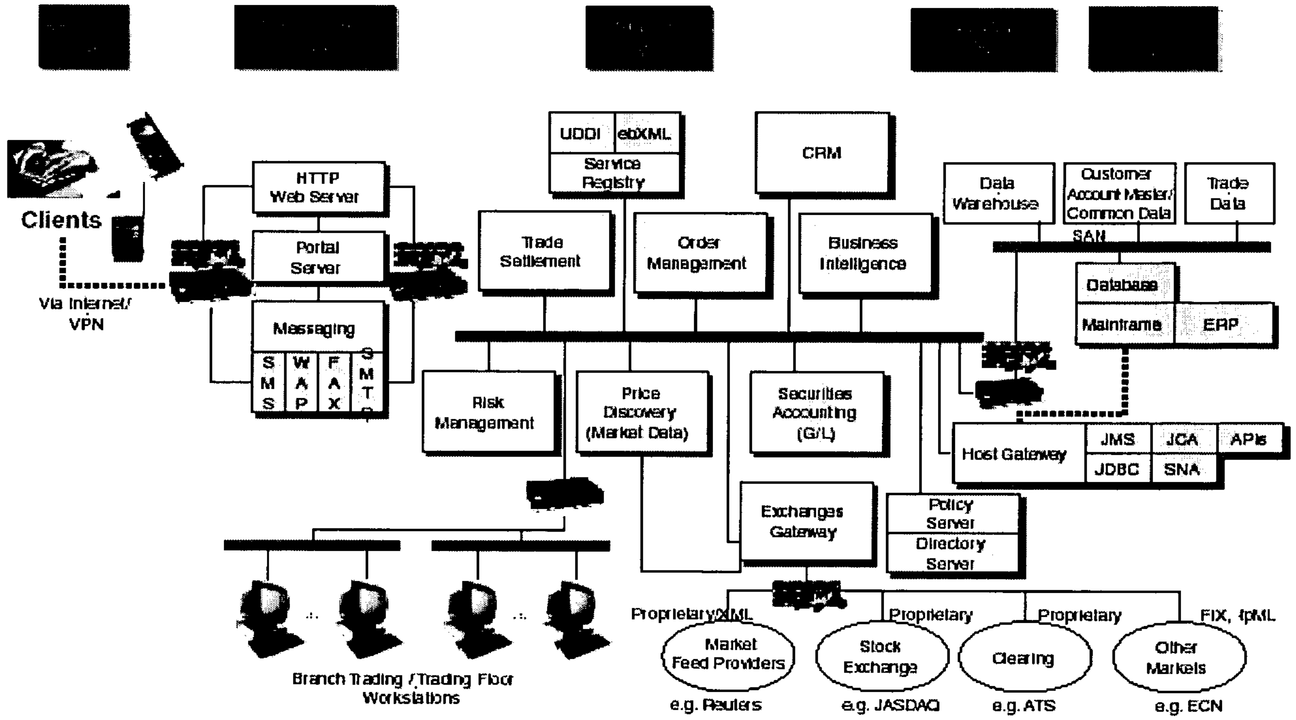

UDDI
ebXML
Service Registry
CRM
Clients
HTTP Web Server
Portal Server
Messaging
S M S
W A P
F A X
S M T P
Via Internet/ VPN
Trade Settlement
Order Management
Business Intelligence
Data Warehouse
Customer Account Master/ Common Data
Trade Data
SAN
Database
Mainframe
ERP
Risk Management
Price Discovery (Market Data)
Securities Accounting (G/L)
Host Gateway
JMS
JCA
APIs
JDBC
SNA
Exchanges Gateway
Policy Server
Directory Server
Proprietary/XML
Proprietary
Proprietary
FIX, fpML
Market Feed Providers
Stock Exchange
Clearing
Other Markets
Branch Trading / Trading Floor Workstations
e.g. Reuters
e.g. JASDAQ
e.g. ATS
e.g. ECN

FIG. 21

Tiers/ Platform Layer	Client Tier	Presentation Tier	Business Tier	Integration Tier	Resource Tier
Application Platform Layer			Order management Trade settlement Risk management Price discovery Securities accounting CRM Business Intelligence	Service Registry	ERP systems Policy Server Directory Server
Virtual Platform Layer		J2EE		SOAP ebXML	Policy Server Directory Server
Upper Platform Layer	Client Browser	Messaging Servers Web Server Portal Server	Application Server		Database Server Policy Server Directory Server
Lower Platform Layer	PDA WAP phone	Solaris OE	Solaris OE	Solaris OE	Policy Server Directory Server
Hardware Platform Layer	PDA WAP phone	Sparc Unix			Mainframe Storage devices/SAN

FIG. 22

"ilities"	Client Tier	Presentation Tier	Business Tier	Integration Tier	Resource Tier
Performance, throughput, and scalability		HTTP-based load balancing for SOAP servlet SOAP/XML cache	Vertical scaling Horizontal scaling	HTTP-based load balancer for Service Registry SOAP/XML cache	Federated Directory Server
Reliability and availability	Reliable and clustered hardware platform	Reliable and clustered hardware platform Clustered messaging servers	Reliable and clustered hardware platform Clustered Application Server	Clustered Service Registry	Master-slave Directory Server for HA Parallel database server Standby database server Reliable and clustered hardware platform
Security	HTTPS VPN gateway	HTTPS VPN gateway	HTTPS	XML security (e.g., DSIG, WS-security)	XML security standards (e.g., SAML, XACML) Trusted Solaris OE
Manageability	System management tools	System management tools	System management tools	System management tools	System management tools
Flexibility		Decoupling presentation from business (e.g., XML for data, HTML for presentation)		Update URL end-point in Service Registry without re-binding run-time (re-compilation)	
Reusability			SOAP-enabled business services	SOAP-enabled business services	SOAP-enabled business services

FIG. 23

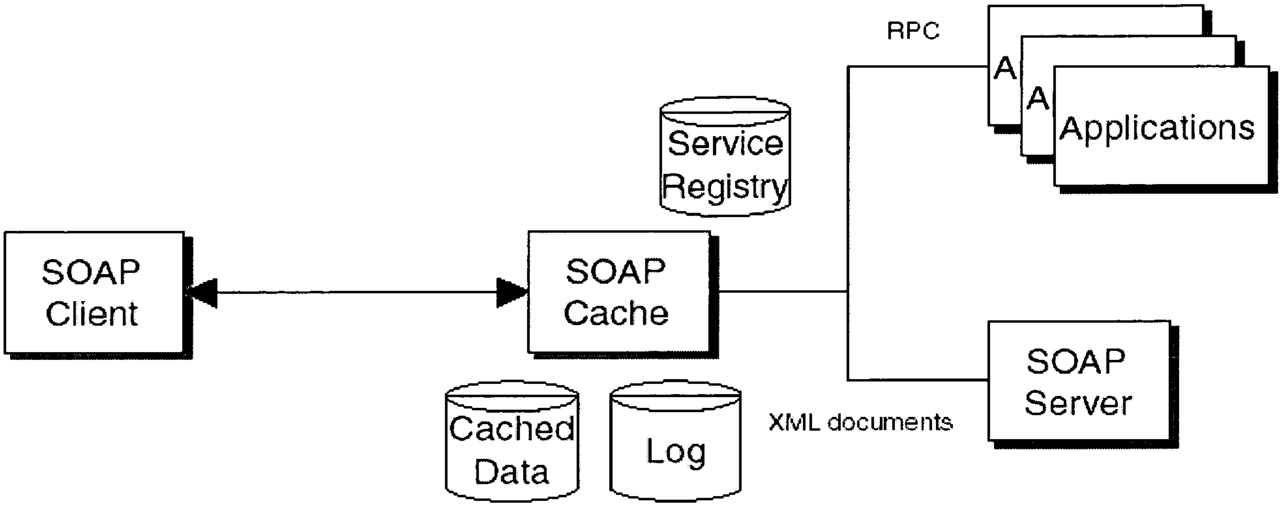

RPC
A
A
Applications
Service Registry
SOAP Client
SOAP Cache
SOAP Server
Cached Data
Log
XML documents

FIG. 24

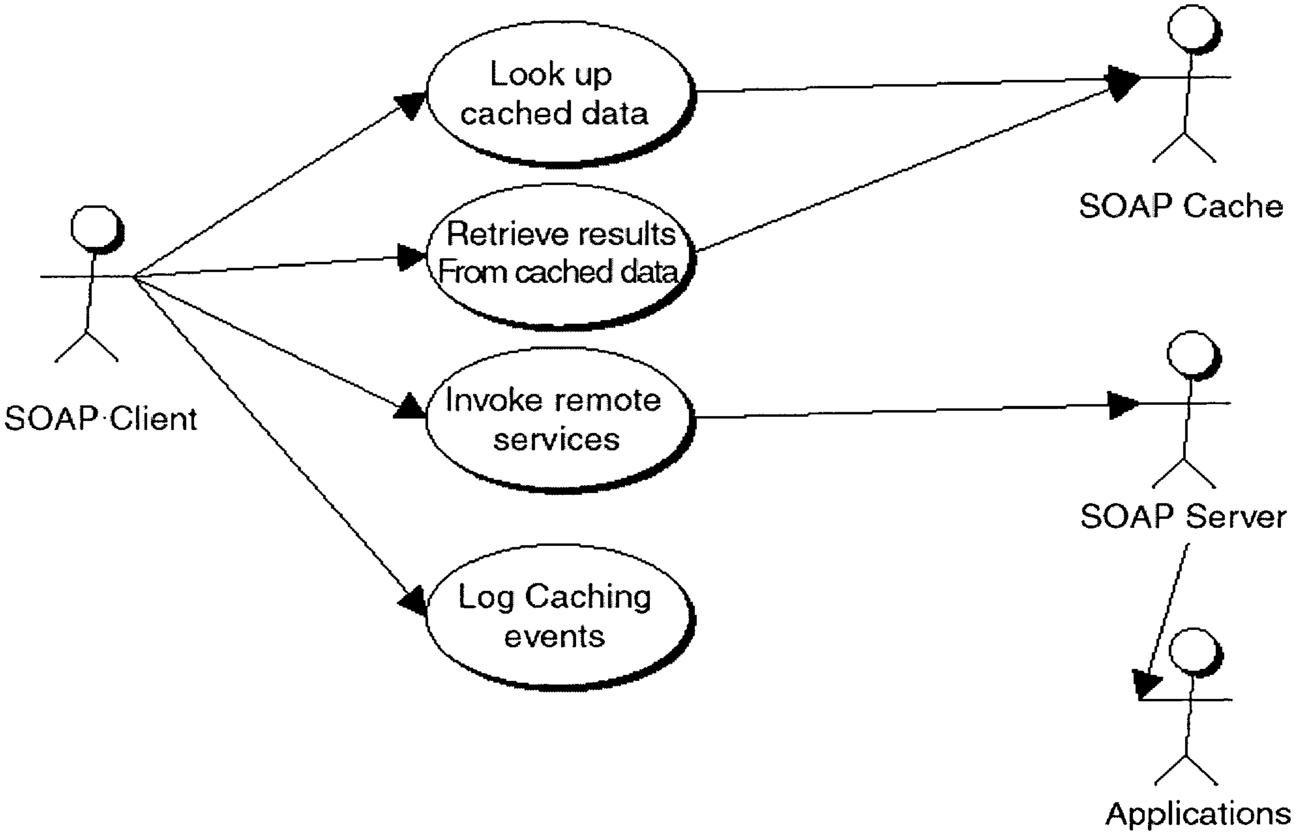

Look up cached data
Retrieve results From cached data
Invoke remote services
Log Caching events
SOAP Client
SOAP Cache
SOAP Server
Applications

FIG. 25

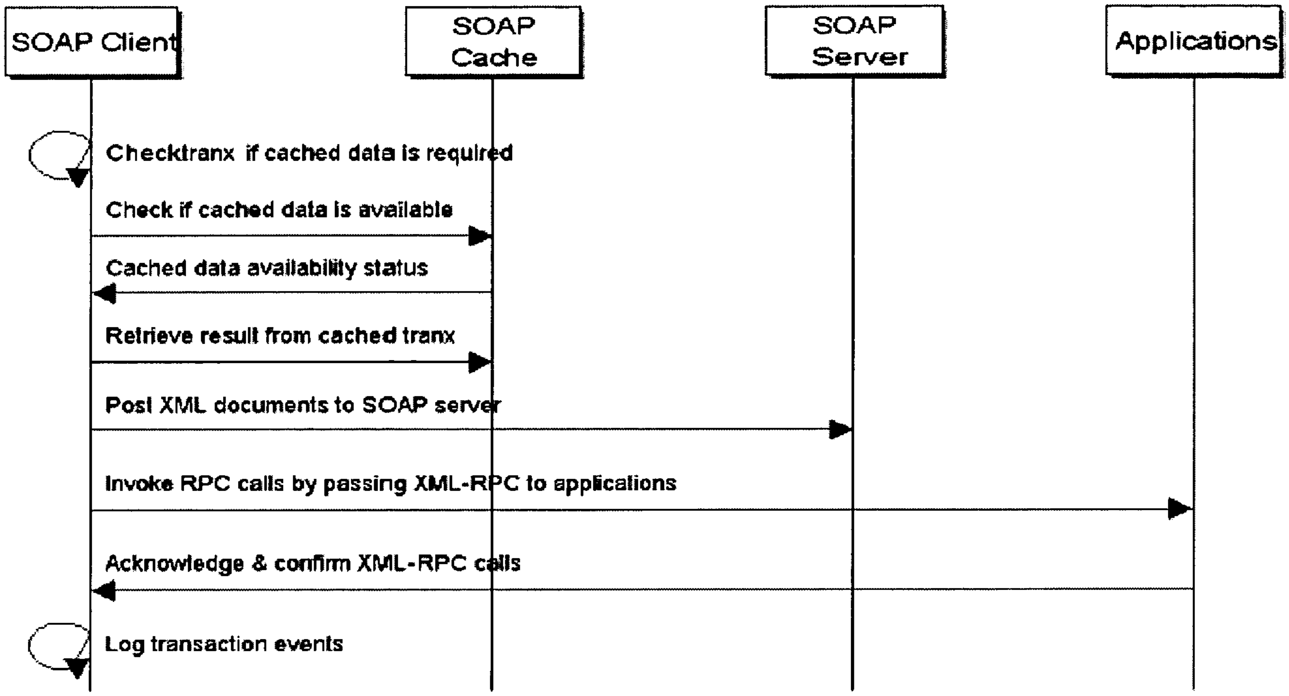

SOAP Client
SOAP Cache
SOAP Server
Applications
Checktranx if cached data is required
Check if cached data is available
Cached data availability status
Retrieve result from cached tranx
Post XML documents to SOAP server
Invoke RPC calls by passing XML-RPC to applications
Acknowledge & confirm XML-RPC calls
Log transaction events

FIG. 26

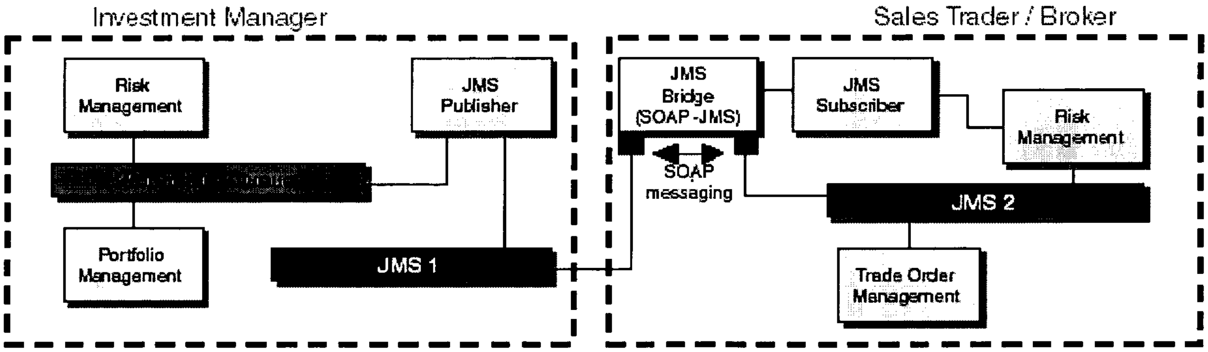

Investment Manager
Risk Management
JMS Publisher
Portfolio Management
JMS 1
Sales Trader / Broker
JMS Bridge (SOAP-JMS)
JMS Subscriber
Risk Management
SOAP messaging
JMS 2
Trade Order Management

FIG. 27

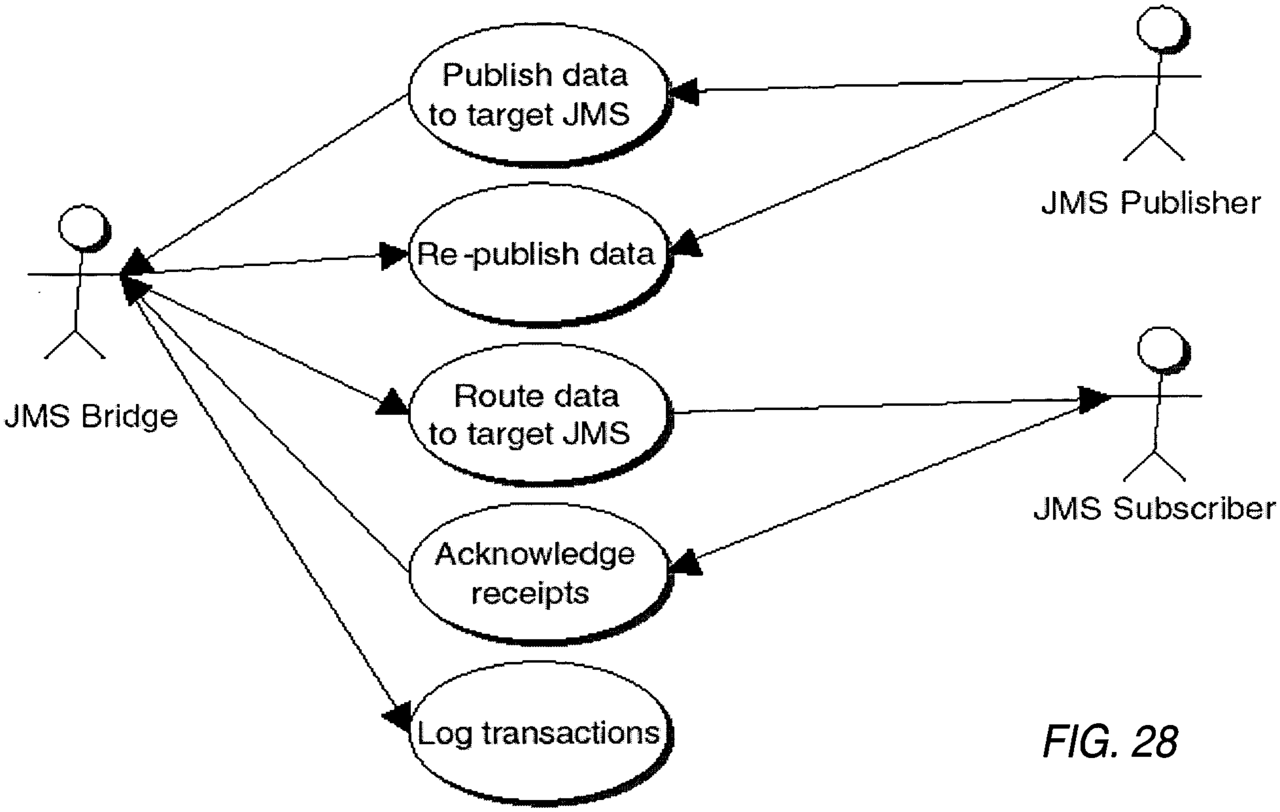

Publish data to target JMS
Re-publish data
Route data to target JMS
Acknowledge receipts
Log transactions
JMS Bridge
JMS Publisher
JMS Subscriber

FIG. 28

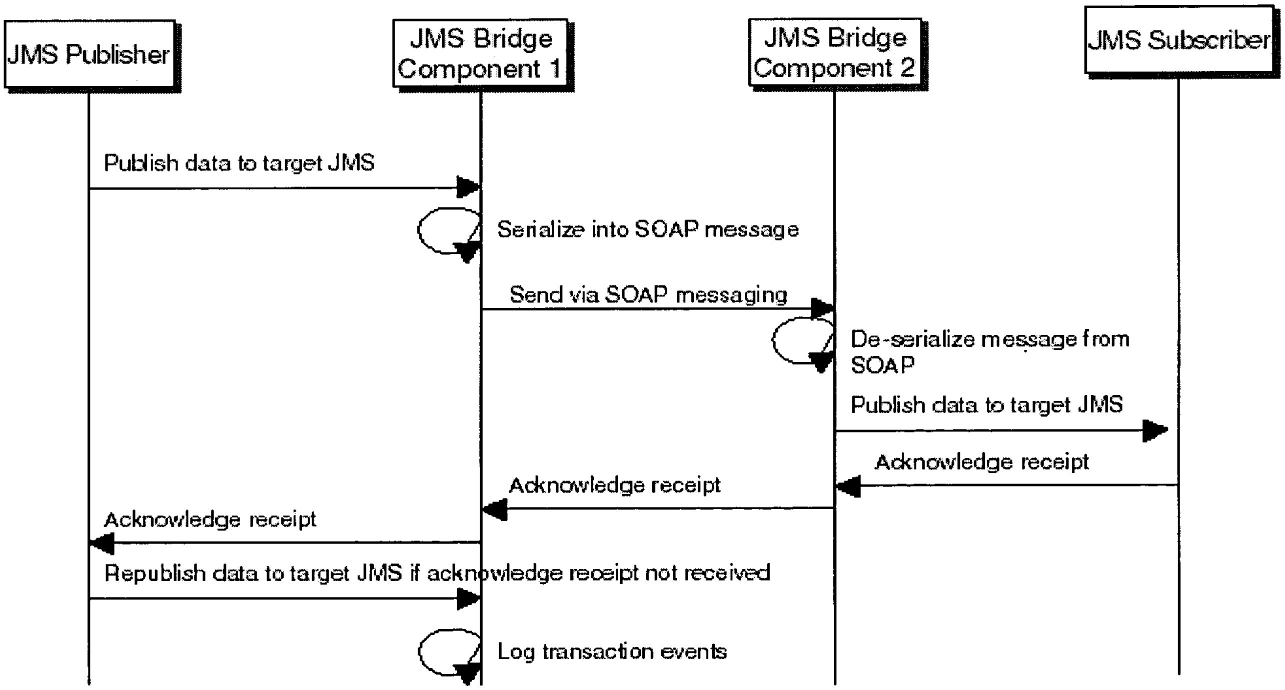

JMS Publisher
JMS Bridge Component 1
JMS Bridge Component 2
JMS Subscriber
Publish data to target JMS
Serialize into SOAP message
Send via SOAP messaging
De-serialize message from SOAP
Publish data to target JMS
Acknowledge receipt
Acknowledge receipt
Acknowledge receipt
Republish data to target JMS if acknowledge receipt not received
Log transaction events

FIG. 29

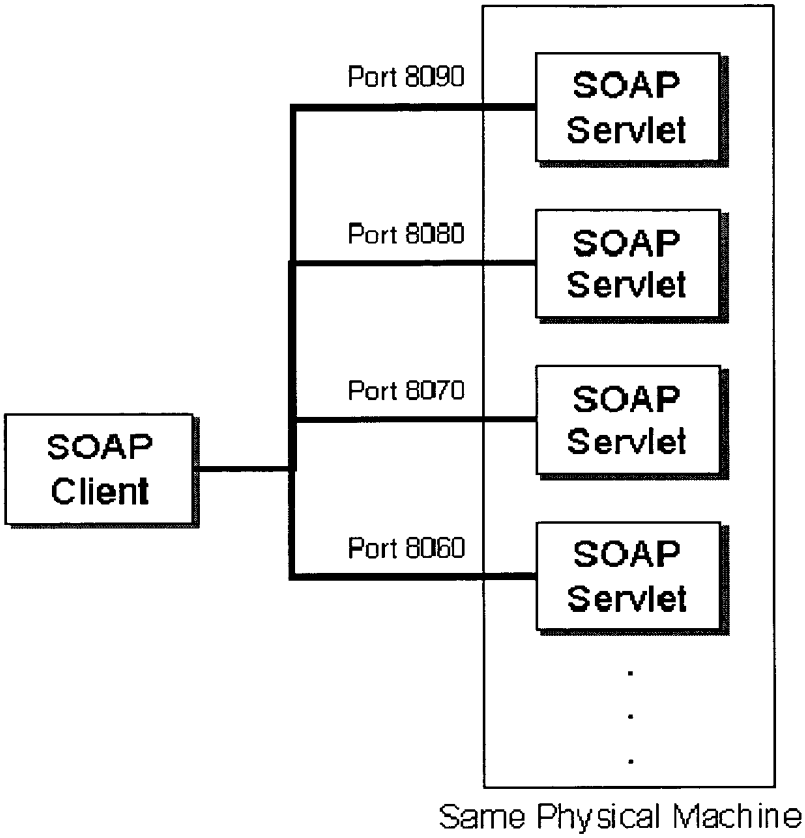

SOAP
Client
Port 8090
Port 8080
Port 8070
Port 8060
SOAP Servlet
SOAP Servlet
SOAP Servlet
SOAP Servlet
Same Physical Machine

FIG. 30

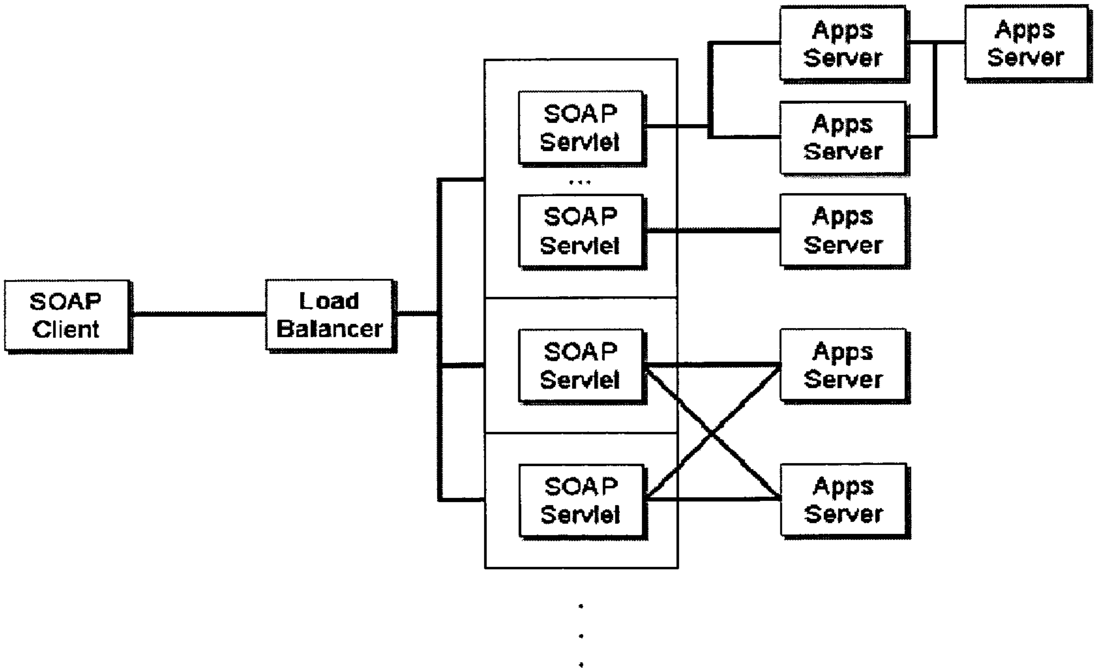

SOAP Client
Load Balancer
SOAP Servlet
SOAP Servlet
SOAP Servlet
SOAP Servlet
Apps Server
Apps Server
Apps Server
Apps Server
Apps Server
Apps Server

FIG. 31

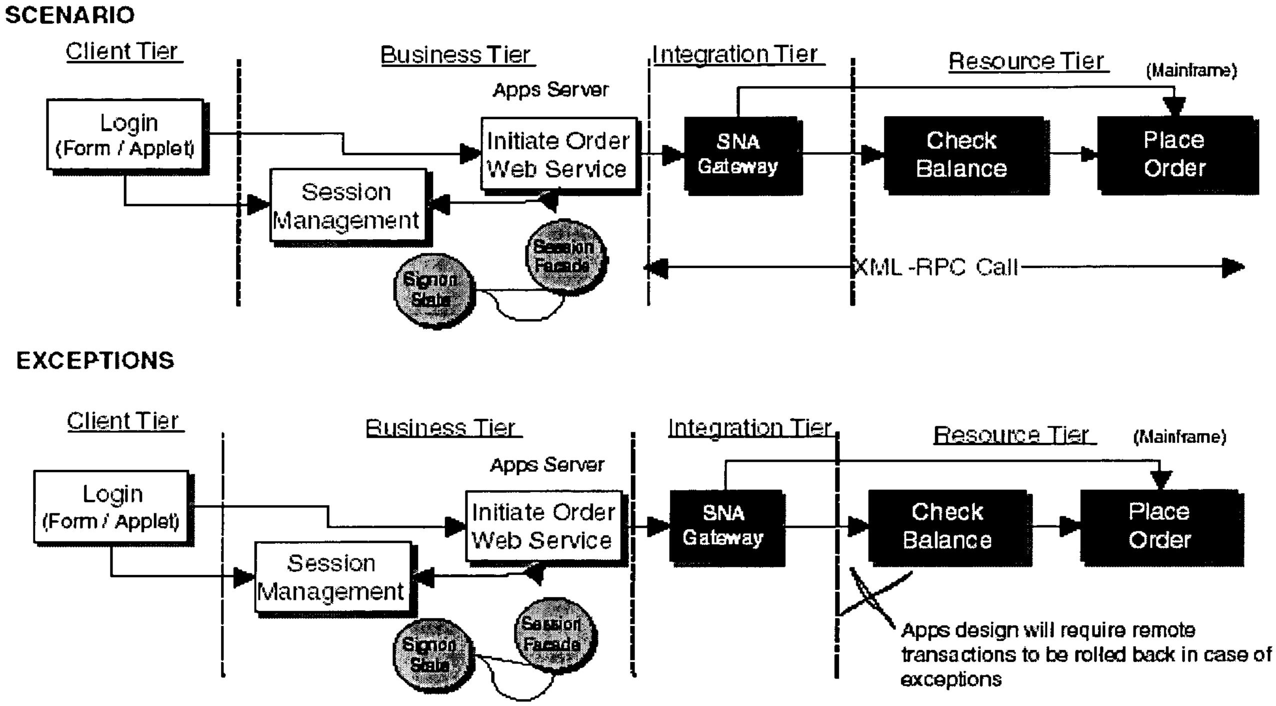

SCENARIO
Client Tier
Business Tier
Integration Tier
Resource Tier
(Mainframe)
Apps Server
Login (Form / Applet)
Initiate Order Web Service
Session Management
Session Facade
Signon State
SNA Gateway
Check Balance
Place Order
XML -RPC Call
EXCEPTIONS
Client Tier
Business Tier
Integration Tier
Resource Tier
(Mainframe)
Apps Server
Login (Form / Applet)
Initiate Order Web Service
Session Management
Session Facade
Signon State
SNA Gateway
Check Balance
Place Order
Apps design will require remote transactions to be rolled back in case of exceptions

FIG. 32

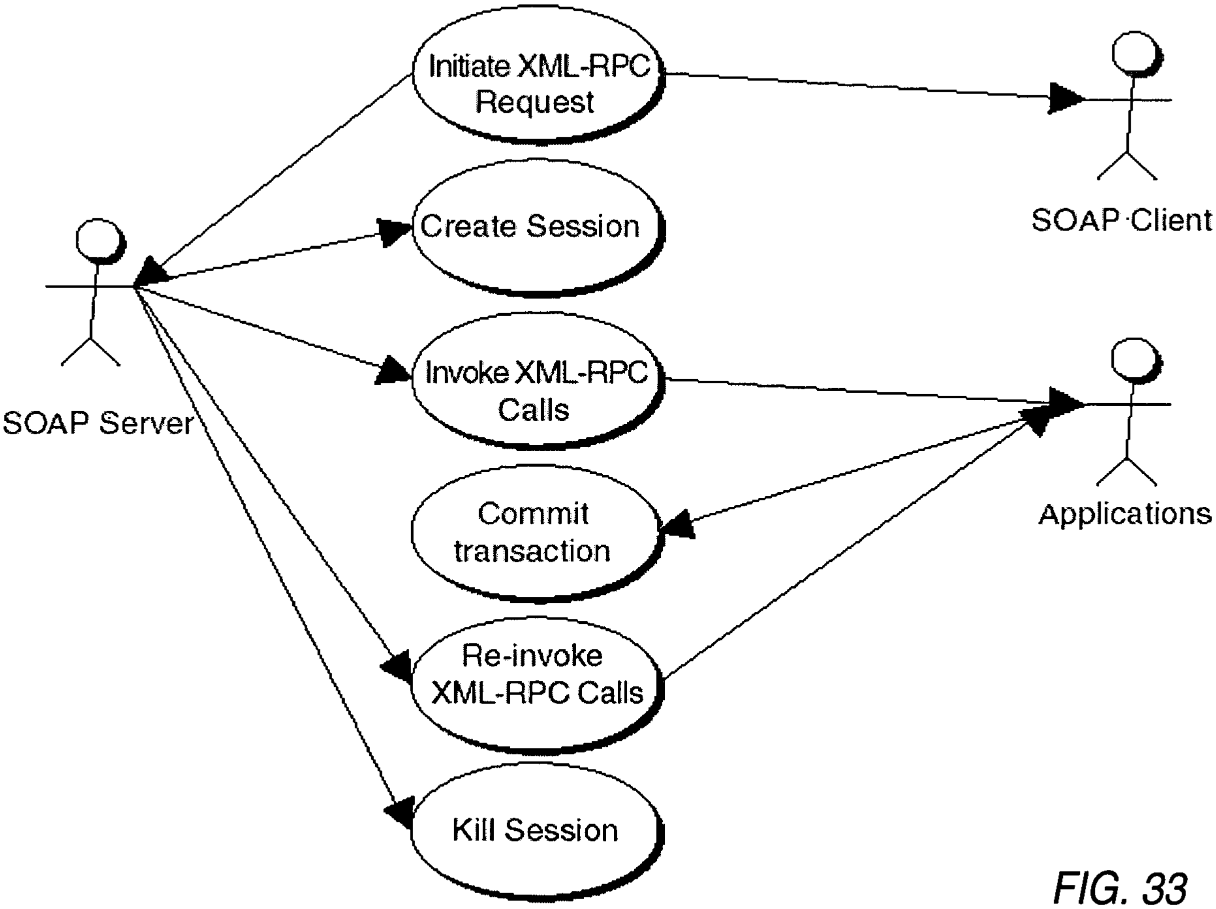

Initiate XML-RPC Request
Create Session
Invoke XML-RPC Calls
Commit transaction
Re-invoke XML-RPC Calls
Kill Session
SOAP Server
SOAP Client
Applications

FIG. 33

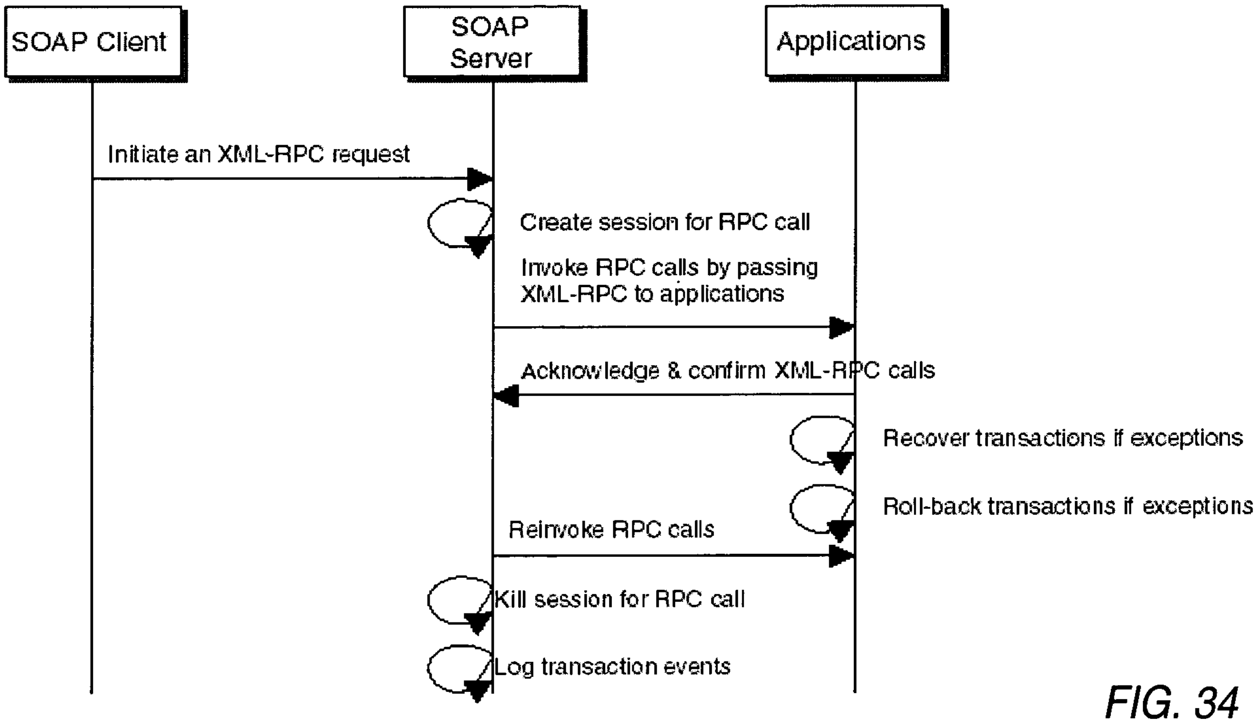

SOAP Client
SOAP Server
Applications
Initiate an XML-RPC request
Create session for RPC call
Invoke RPC calls by passing XML-RPC to applications
Acknowledge & confirm XML-RPC calls
Recover transactions if exceptions
Roll-back transactions if exceptions
Reinvoke RPC calls
Kill session for RPC call
Log transaction events

FIG. 34

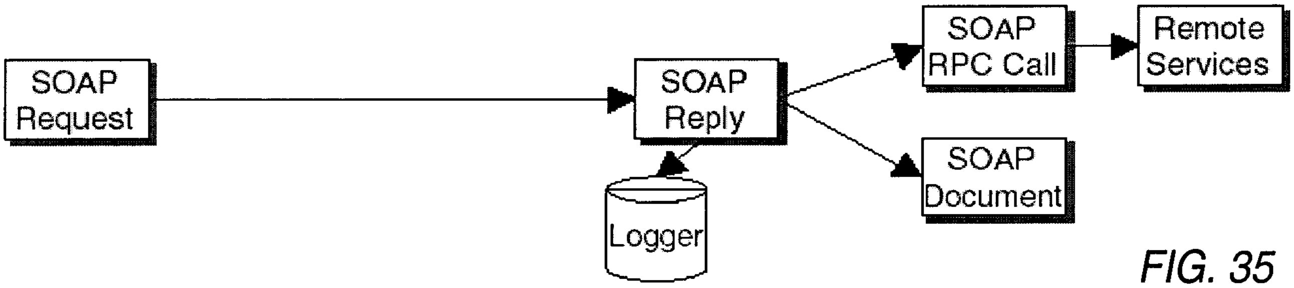


FIG. 35

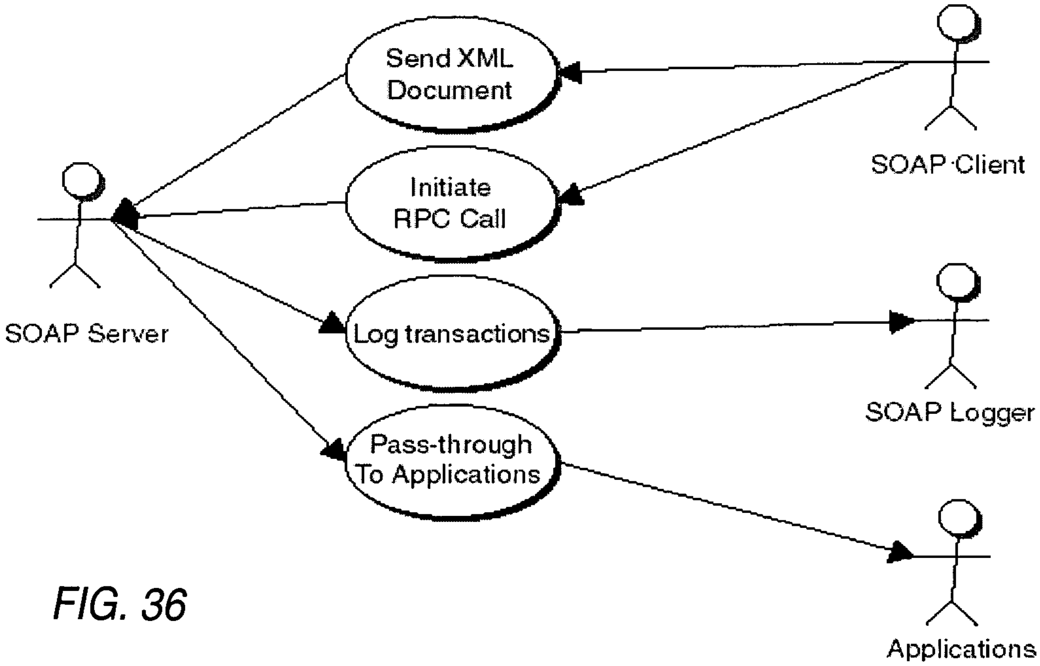


FIG. 36

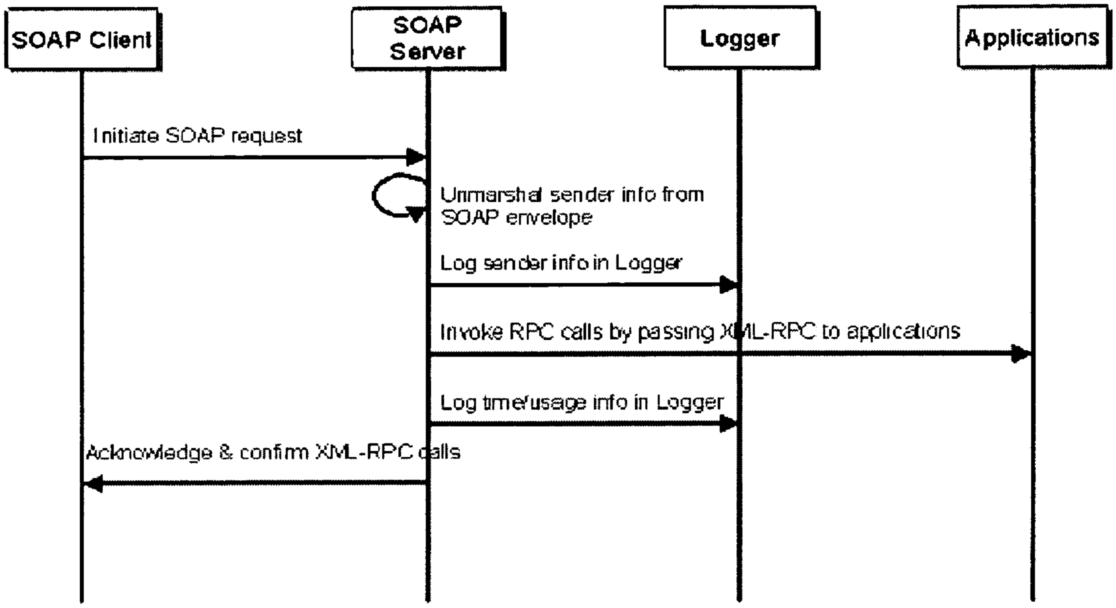

SOAP Client
SOAP Server
Logger
Applications
Initiate SOAP request
Unmarshal sender info from SOAP envelope
Log sender info in Logger
Invoke RPC calls by passing XML-RPC to applications
Log time/usage info in Logger
Acknowledge & confirm XML-RPC calls

FIG. 37

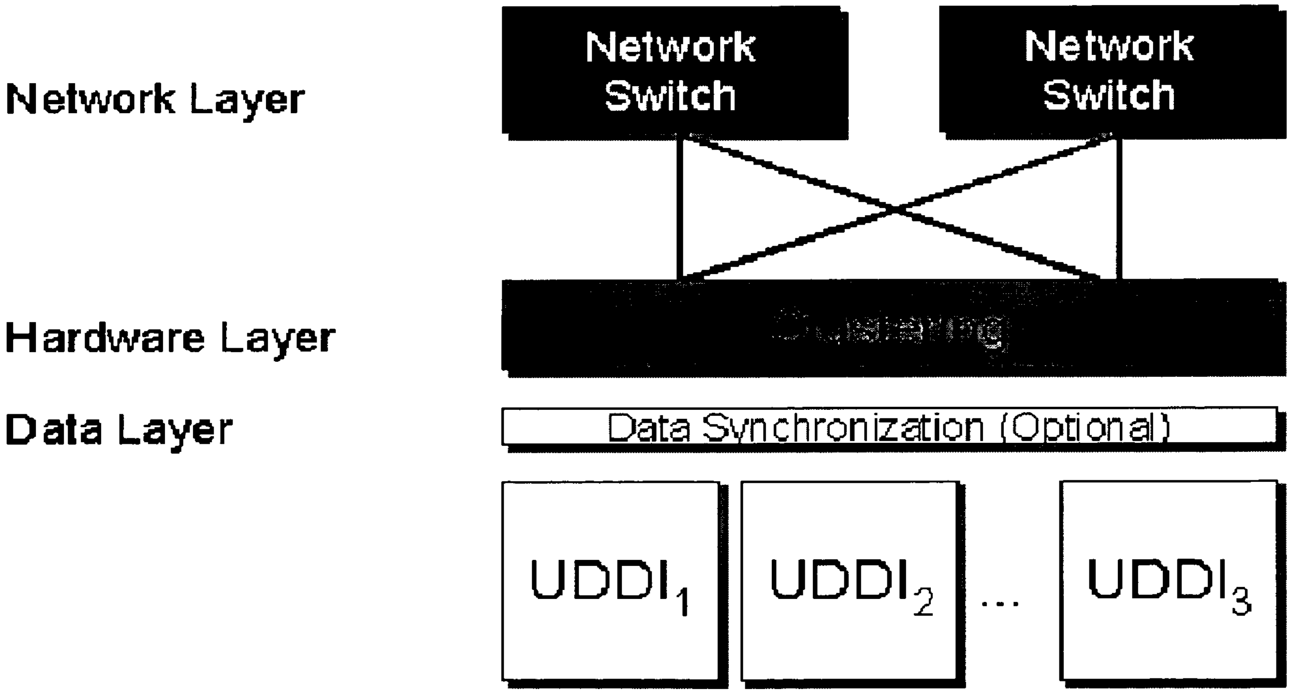
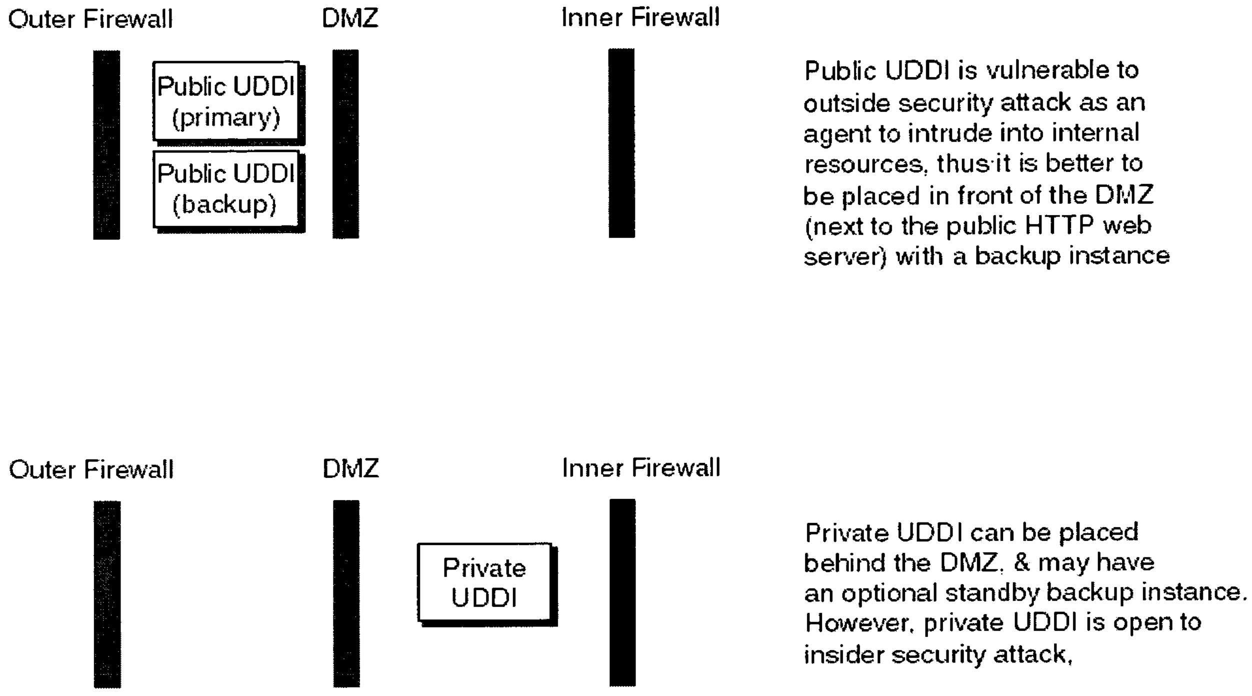

Network Layer
Hardware Layer
Data Layer
Network Switch
Network Switch
Clustering
Data Synchronization (Optional)
UDDI$_1$
UDDI$_2$
...
UDDI$_3$

FIG. 38



Public UDDI is vulnerable to outside security attack as an agent to intrude into internal resources, thus it is better to be placed in front of the DMZ (next to the public HTTP web server) with a backup instance

Private UDDI can be placed behind the DMZ, & may have an optional standby backup instance. However, private UDDI is open to insider security attack,

FIG. 39

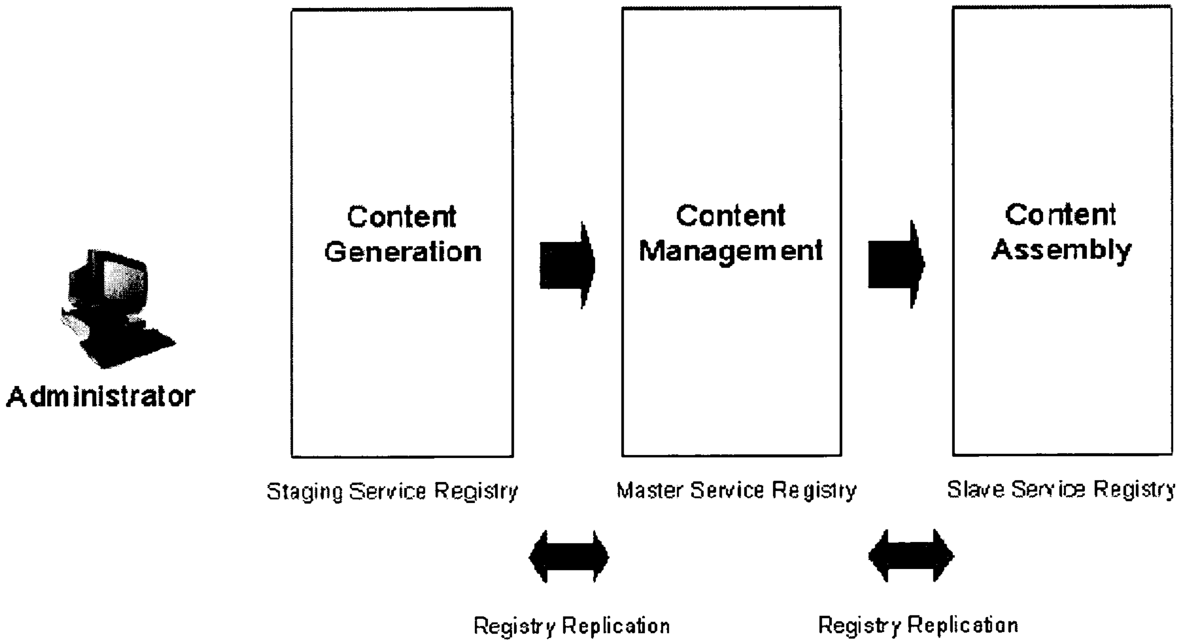


FIG. 40

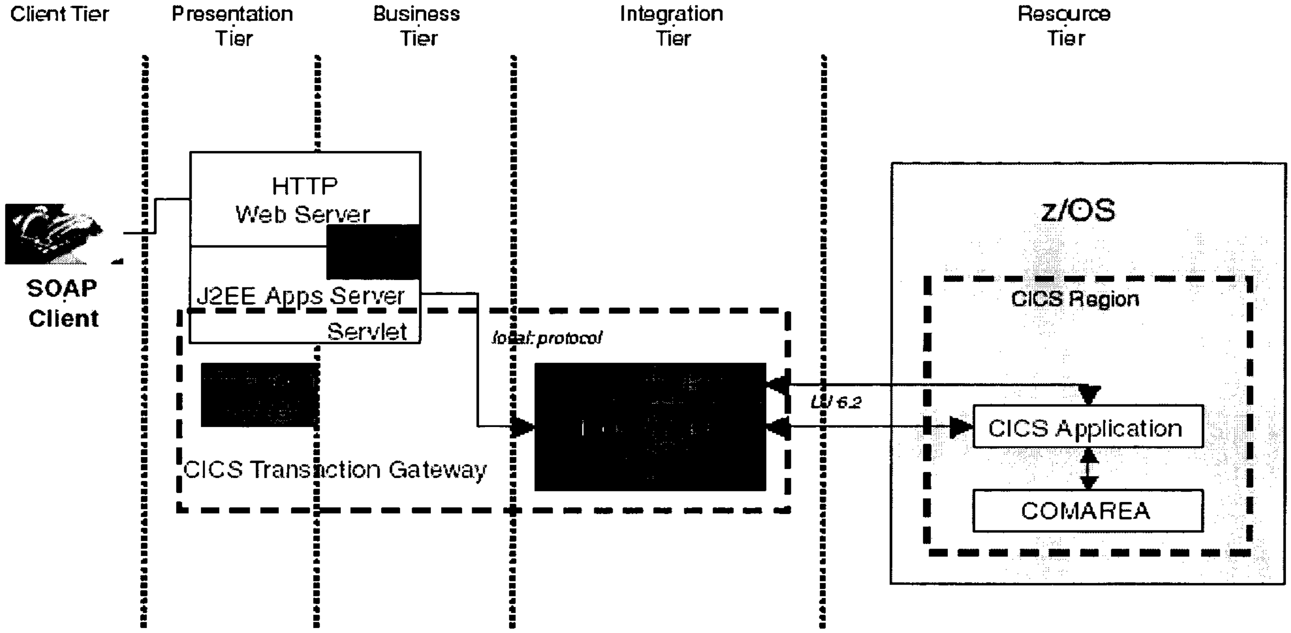

Client Tier
Presentation Tier
Business Tier
Integration Tier
Resource Tier
SOAP Client
HTTP Web Server
J2EE Apps Server
Servlet
local protocol
CICS Transaction Gateway
LU 6.2
z/OS
CICS Region
CICS Application
COMAREA

FIG. 41

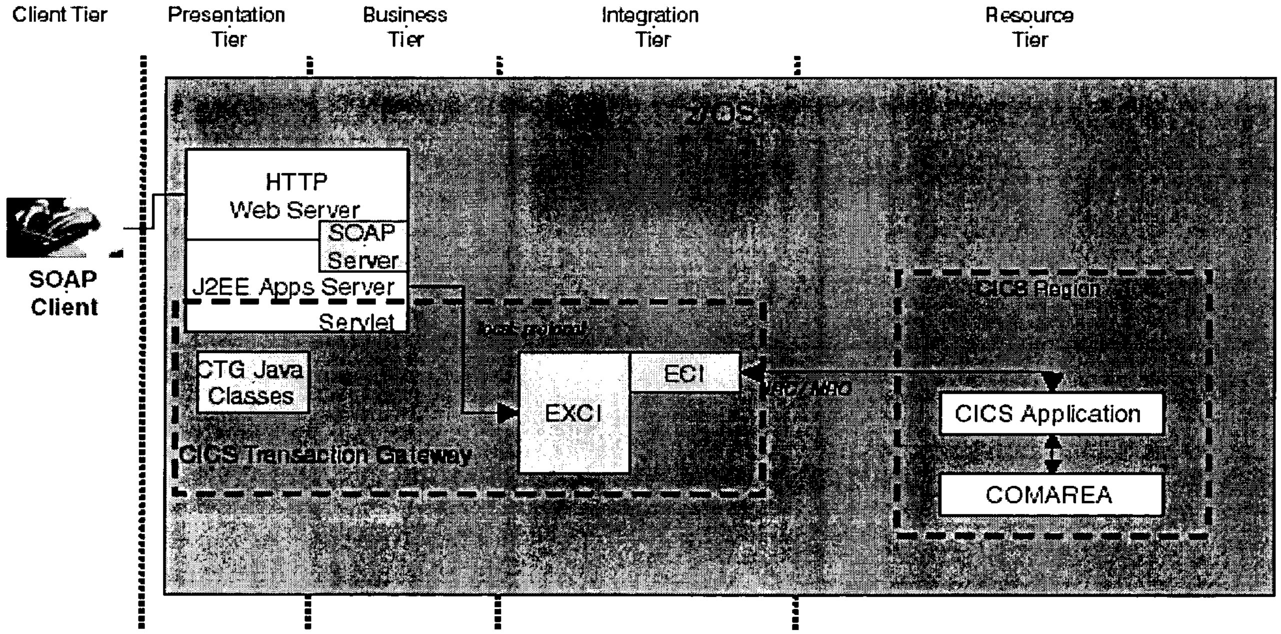

Client Tier
Presentation Tier
Business Tier
Integration Tier
Resource Tier
z/OS
SOAP Client
HTTP Web Server
SOAP Server
J2EE Apps Server
Servlet
local protocol
CTG Java Classes
EXCI
ECI
CICS Transaction Gateway
CICS Region
CICS Application
COMAREA

FIG. 42

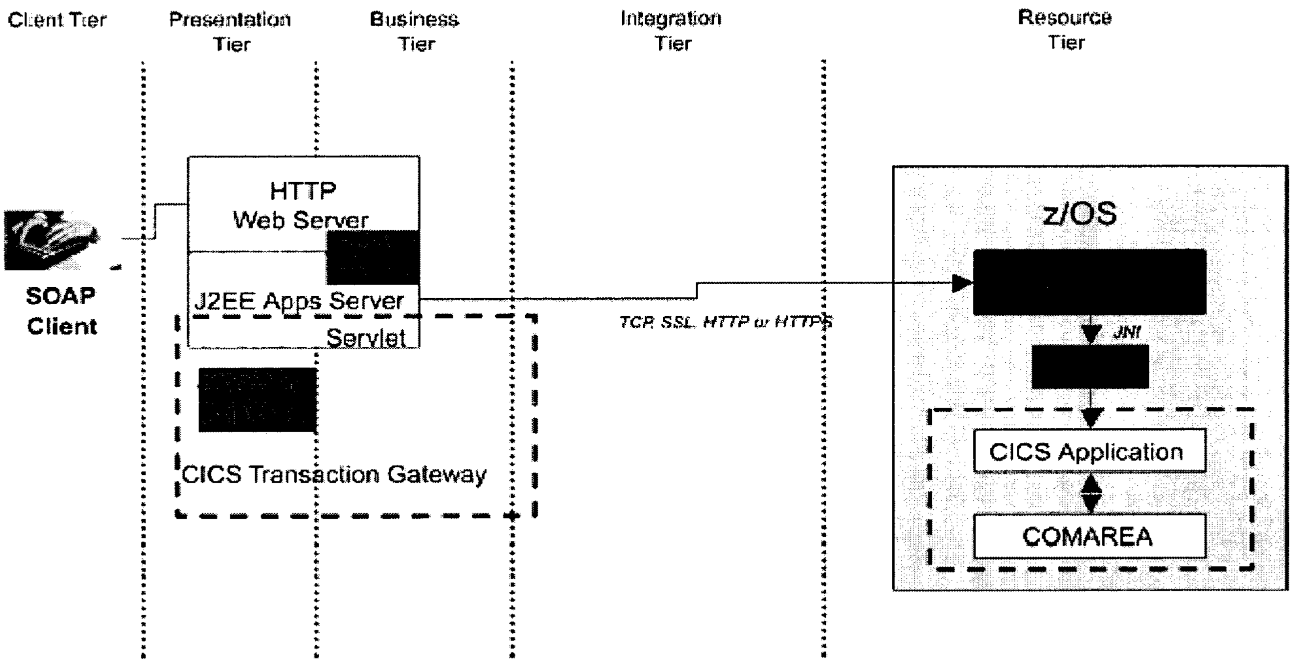

Client Tier
Presentation Tier
Business Tier
Integration Tier
Resource Tier
SOAP Client
HTTP Web Server
J2EE Apps Server
Servlet
CICS Transaction Gateway
TCP, SSL, HTTP or HTTPS
z/OS
JNI
CICS Application
COMAREA

FIG. 43

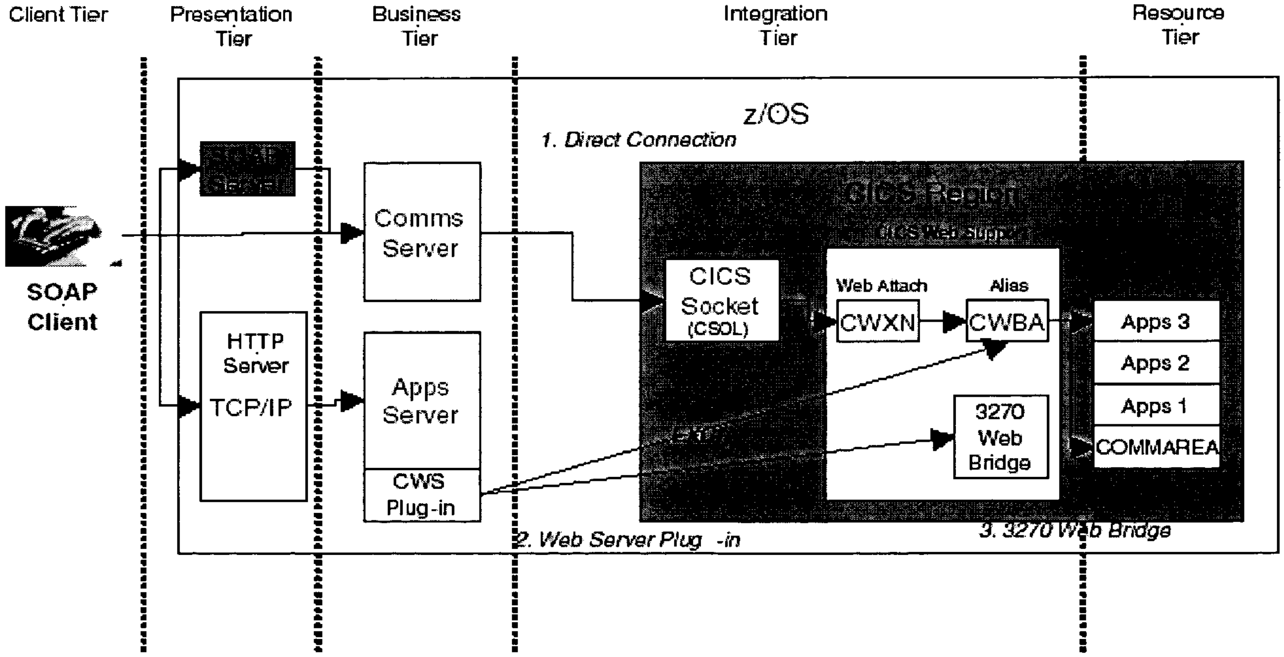

Client Tier
Presentation Tier
Business Tier
Integration Tier
Resource Tier
z/OS
1. Direct Connection
SOAP Client
SOAP Server
Comms Server
HTTP Server
TCP/IP
Apps Server
CWS Plug-in
CICS Region
CICS Web Support
CICS Socket (CSOL)
Web Attach
CWXN
Alias
CWBA
3270 Web Bridge
Apps 3
Apps 2
Apps 1
COMMAREA
2. Web Server Plug -in
3. 3270 Web Bridge

FIG. 44

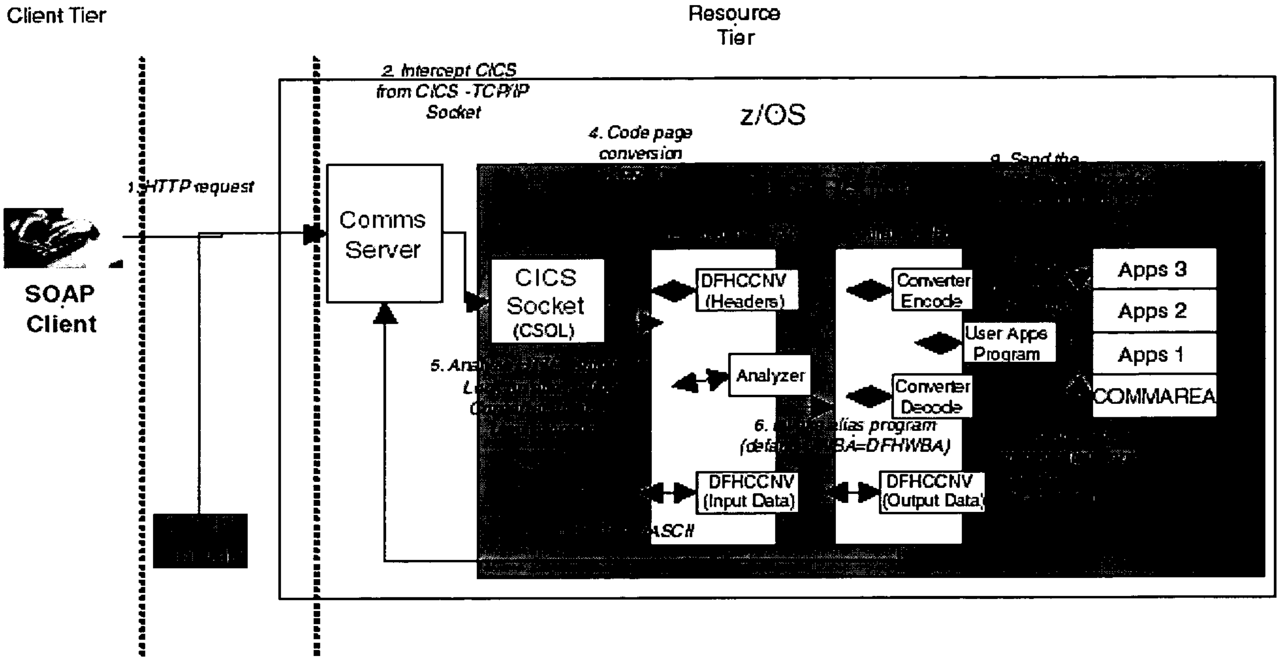

Client Tier
Resource Tier
2. Intercept CICS from CICS -TCP/IP Socket
z/OS
4. Code page conversion
9. Send the
1. HTTP request
Comms Server
CICS Socket (CSOL)
DFHCCNV (Headers)
Converter Encode
Apps 3
Apps 2
Apps 1
COMMAREA
User Apps Program
SOAP Client
5. Ana
Analyzer
Converter Decode
6. alias program
(defa BA=DFHWBA)
DFHCCNV (Input Data)
DFHCCNV (Output Data)
ASCII

FIG. 45

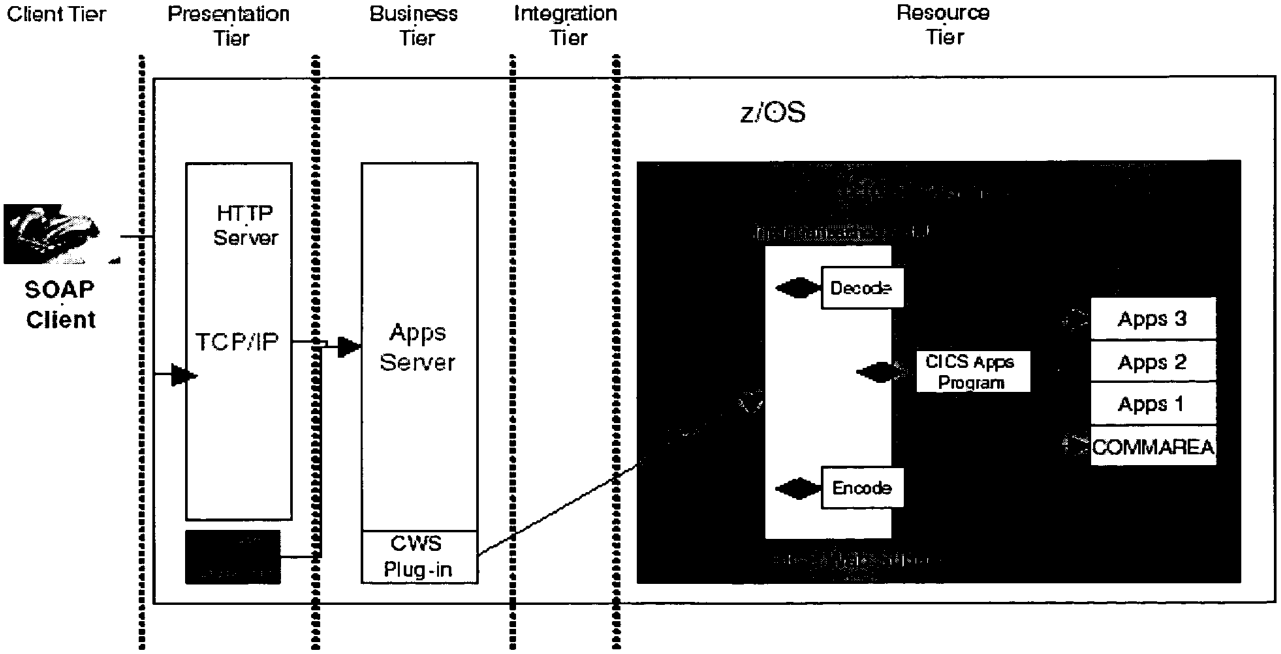

Client Tier
Presentation Tier
Business Tier
Integration Tier
Resource Tier
z/OS
HTTP Server
TCP/IP
SOAP Client
Apps Server
CWS Plug-in
Decode
CICS Apps Program
Encode
Apps 3
Apps 2
Apps 1
COMMAREA

FIG. 46

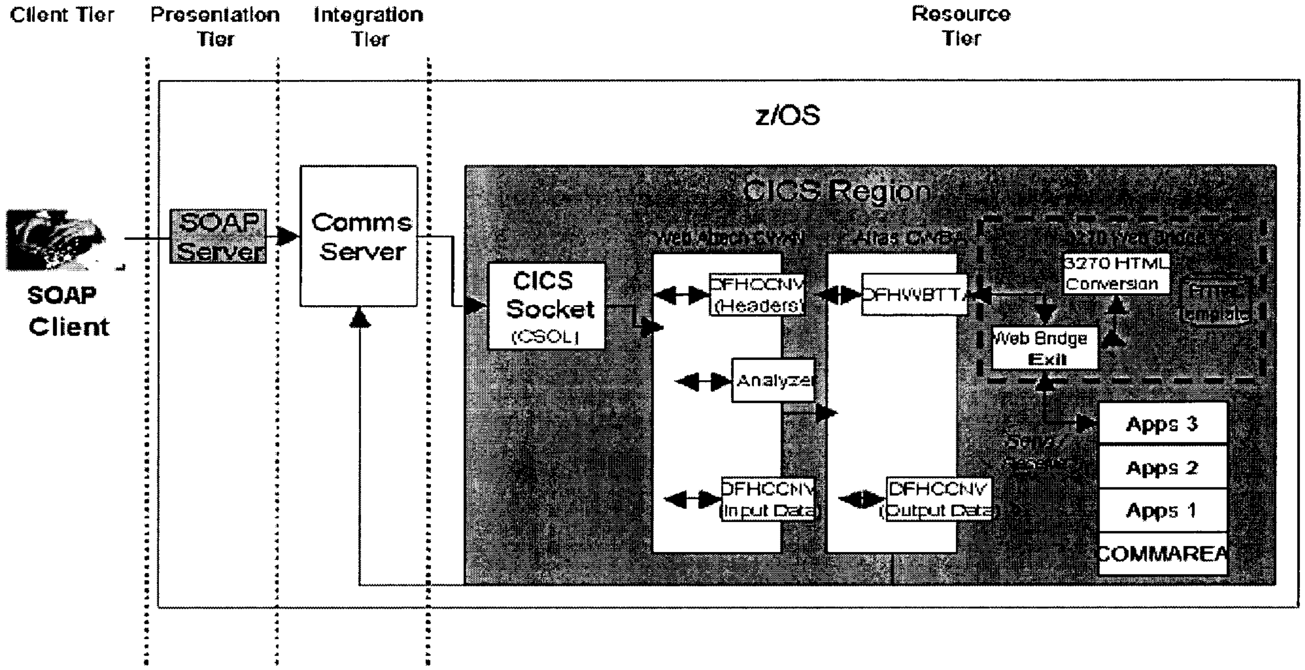

Client Tier
Presentation Tier
Integration Tier
Resource Tier
z/OS
SOAP Client
SOAP Server
Comms Server
CICS Region
CICS Socket (CSOL)
Web Attach CWXN
Alias CWBA
DFHCCNV (Headers)
Analyzer
DFHCCNV (Input Data)
DFHWBTTA
DFHCCNV (Output Data)
3270 Web Bridge
3270 HTML Conversion
HTML template
Web Bridge Exit
Send / Receive
Apps 3
Apps 2
Apps 1
COMMAREA

FIG. 47

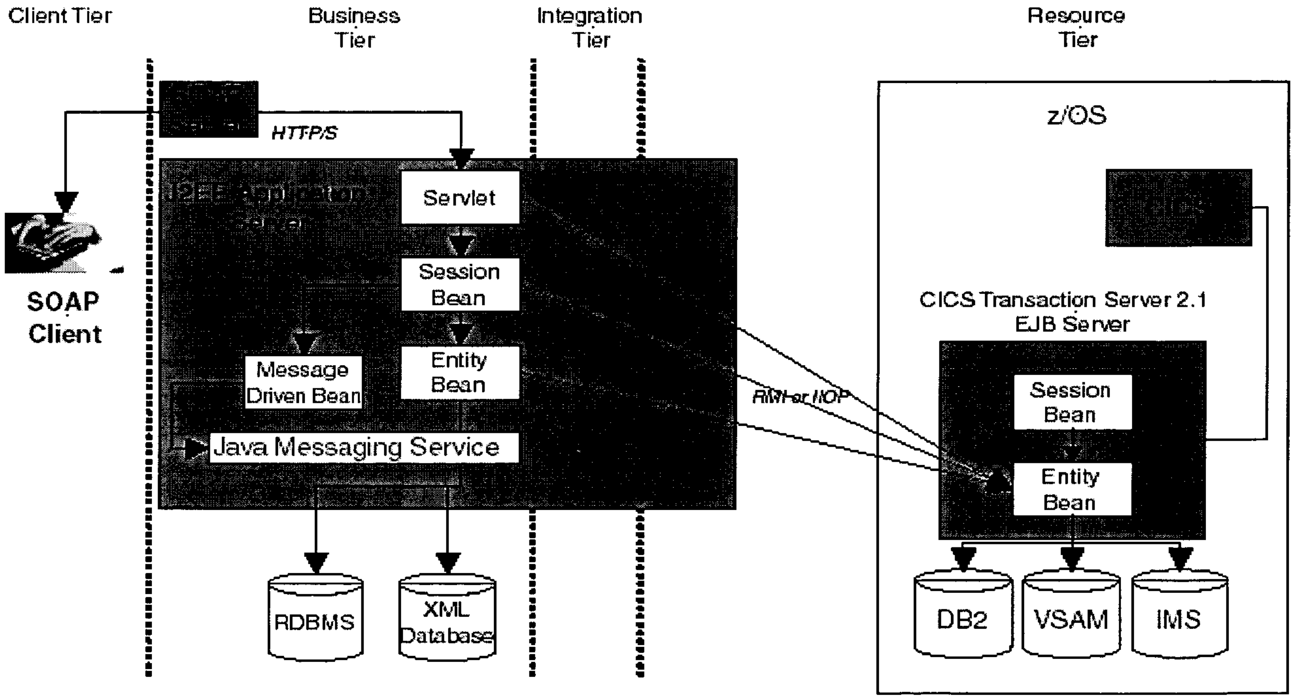

Client Tier
Business Tier
Integration Tier
Resource Tier
HTTP/S
SOAP Client
J2EE Application Server
Servlet
Session Bean
Entity Bean
Message Driven Bean
Java Messaging Service
RDBMS
XML Database
RMI or IIOP
z/OS
CICS
CICS Transaction Server 2.1 EJB Server
Session Bean
Entity Bean
DB2
VSAM
IMS

FIG. 48

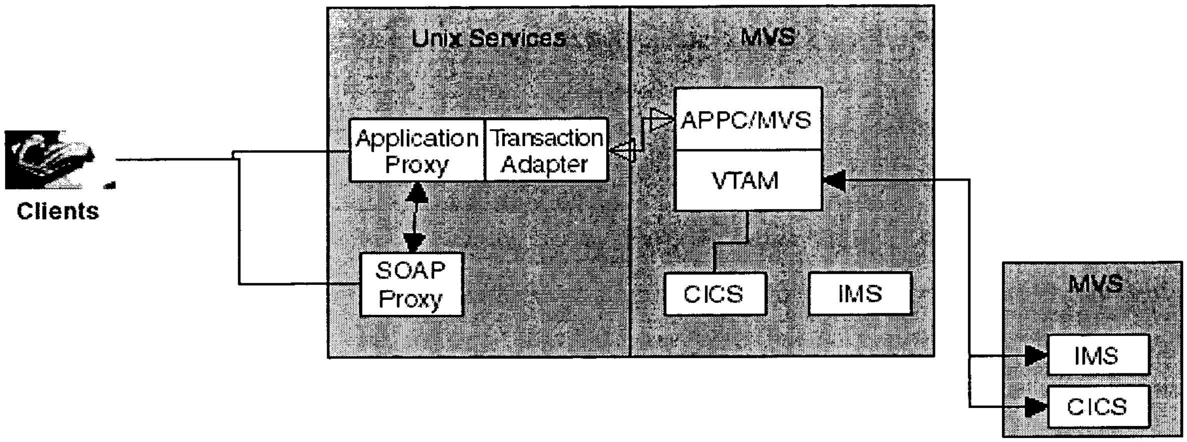


FIG. 49

Technology Approach	Business Tier (Application Server)	Integration Tier	Resource Tier (Back-End Legacy System)
CICS Transaction Gateway		CICS Transaction Gateway—use of ECI, EPI, and ESI calls	
CICS Web Support			CICS Web Support—using CWS to Web-enable 3270-based CICS applications
Java	Enterprise Java Beans—abstracting business functionality from legacy systems	Java Connector Architecture—standardizing connectors to legacy systems	CICS EJB Server—EJB container to support EJB
SOAP Proxy on Mainframe		Forte Transaction Adapter—building Application Proxy for back-end resources	Forte Transaction Adapter—server side for APPC conversation

FIG. 50

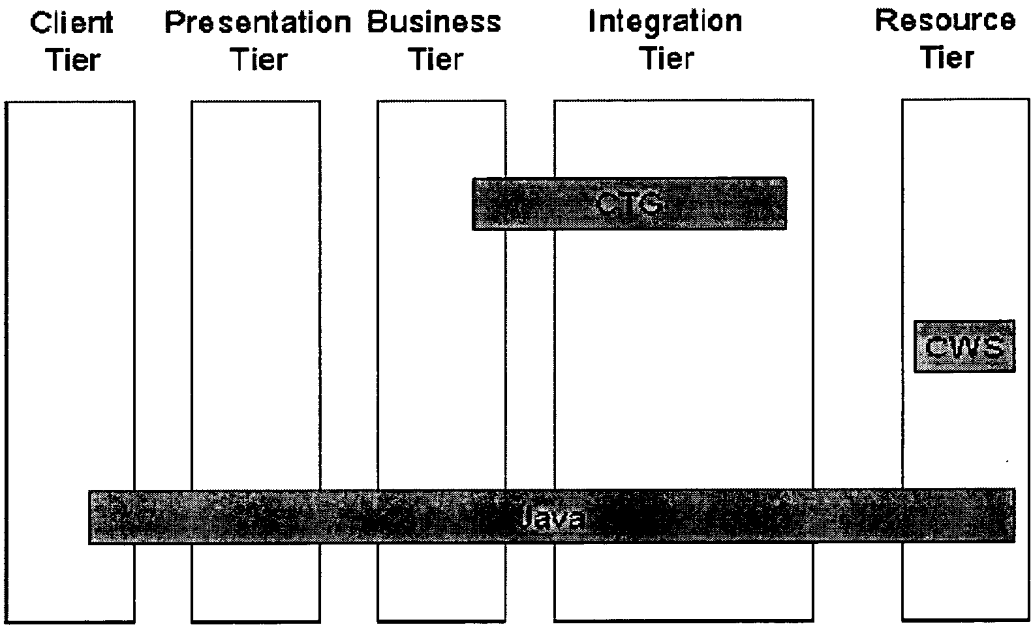

Client Tier
Presentation Tier
Business Tier
Integration Tier
Resource Tier
CTG
CWS
Java

FIG. 51

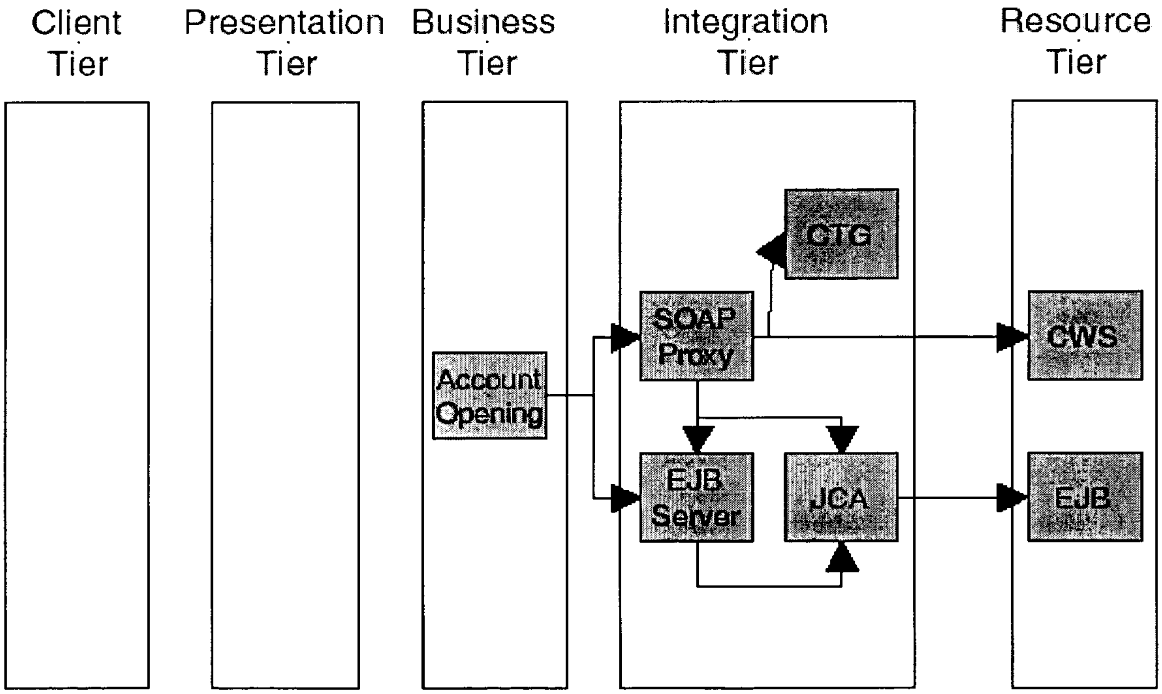

Client Tier
Presentation Tier
Business Tier
Integration Tier
Resource Tier
CTG
SOAP Proxy
CWS
Account Opening
EJB Server
JCA
EJB

FIG. 52

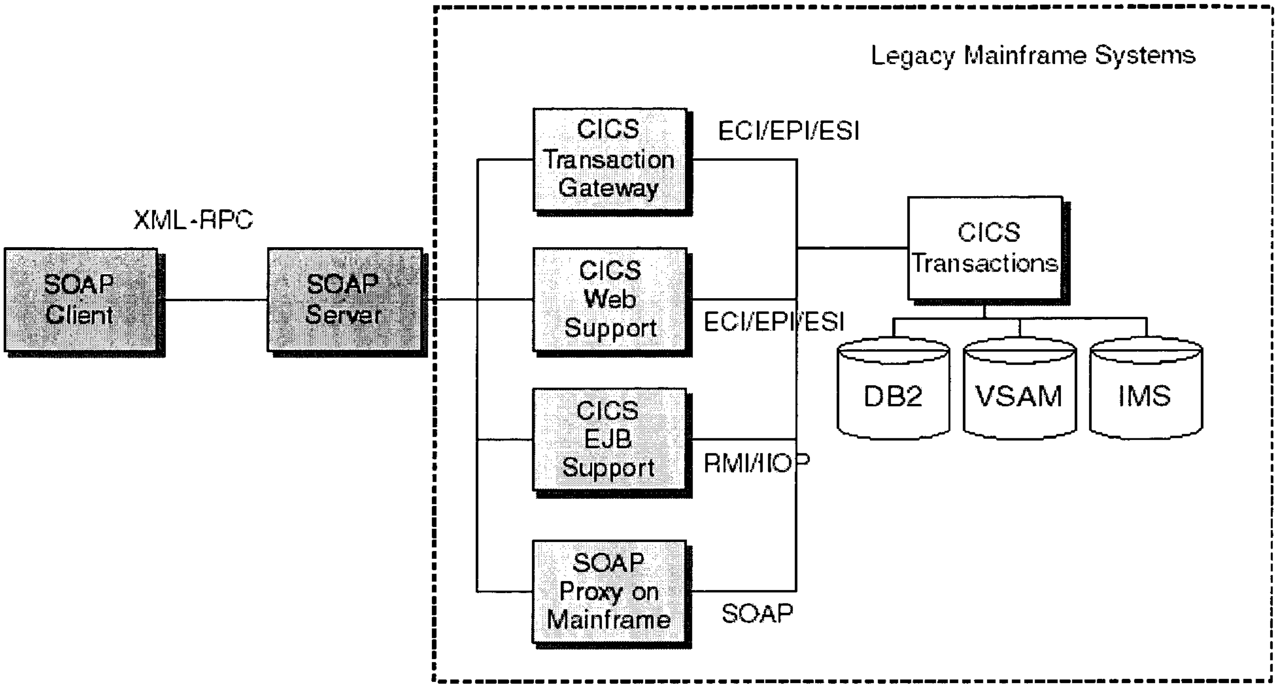

Legacy Mainframe Systems
XML-RPC
SOAP Client
SOAP Server
CICS Transaction Gateway
ECI/EPI/ESI
CICS Web Support
ECI/EPI/ESI
CICS EJB Support
RMI/IIOP
SOAP Proxy on Mainframe
SOAP
CICS Transactions
DB2
VSAM
IMS

FIG. 53

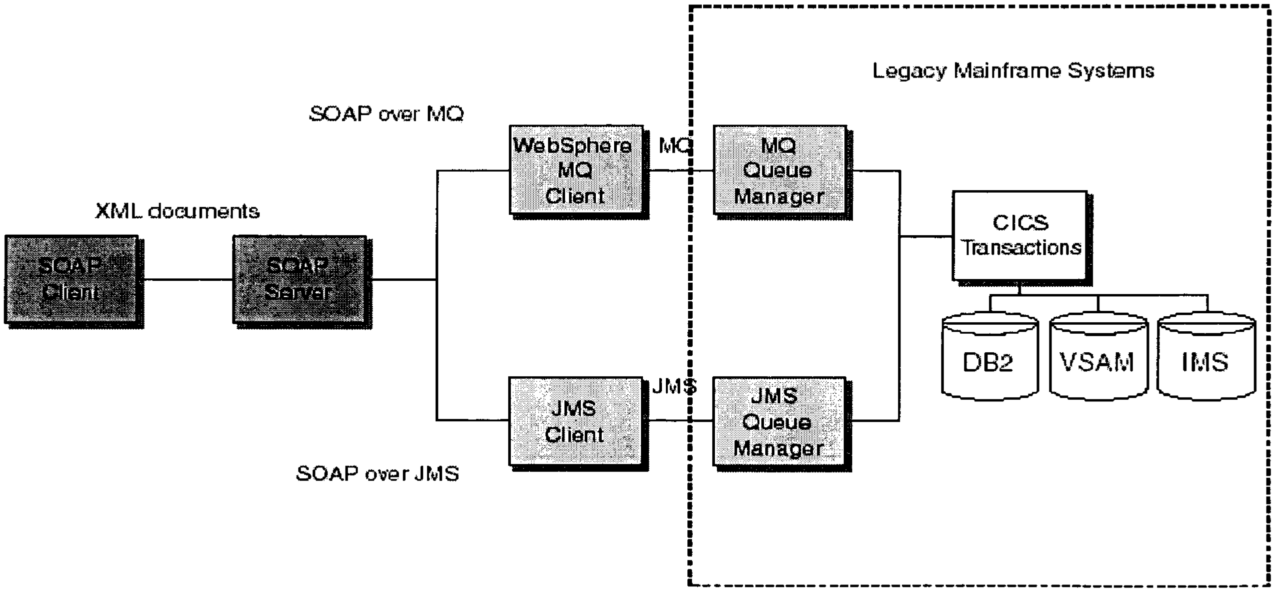

Legacy Mainframe Systems
SOAP over MQ
XML documents
SOAP Client
SOAP Server
WebSphere MQ Client
MQ
MQ Queue Manager
CICS Transactions
DB2
VSAM
IMS
JMS Client
JMS
JMS Queue Manager
SOAP over JMS

FIG. 54

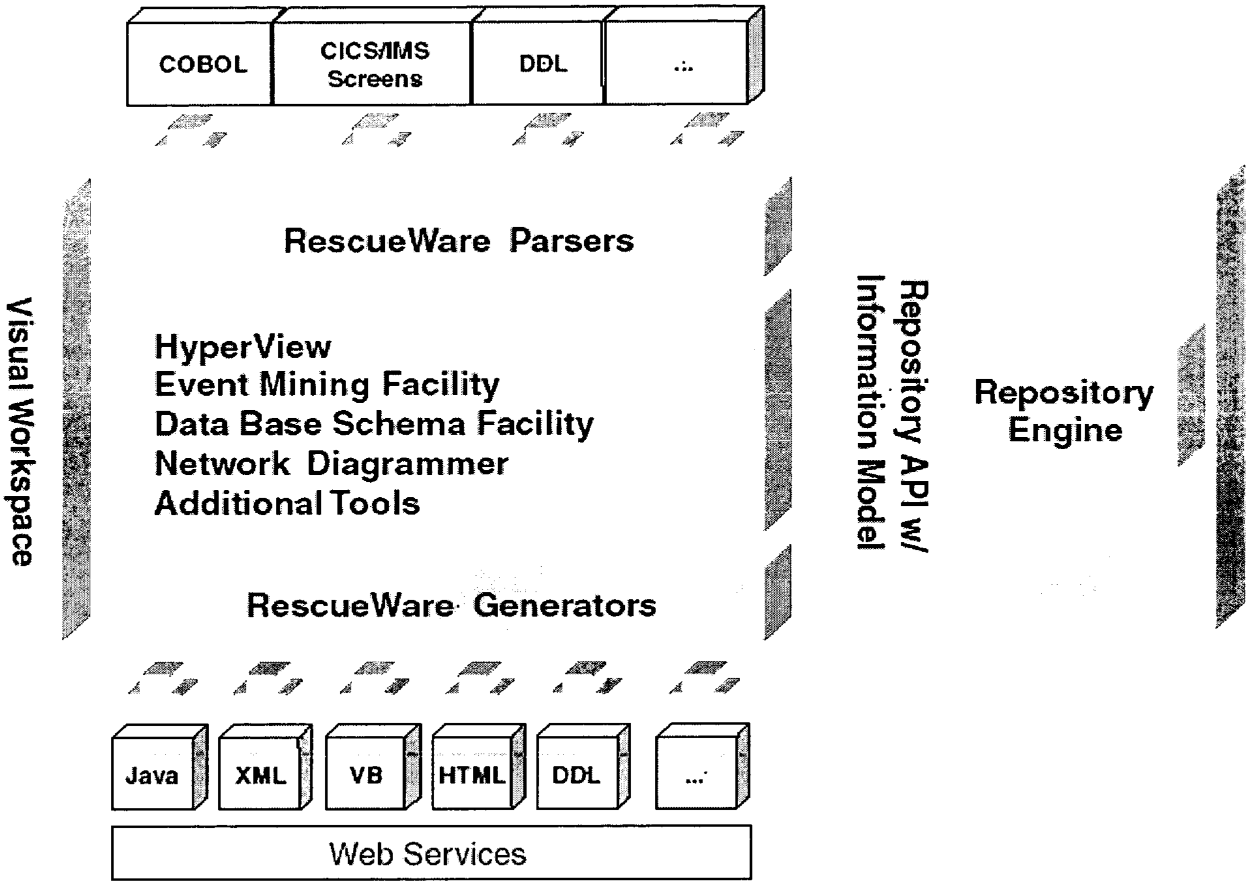

COBOL
CICS/IMS Screens
DDL
.:.
RescueWare Parsers
Visual Workspace
HyperView
Event Mining Facility
Data Base Schema Facility
Network Diagrammer
Additional Tools
Repository API w/ Information Model
Repository Engine
RescueWare Generators
Java
XML
VB
HTML
DDL
...
Web Services

FIG. 55

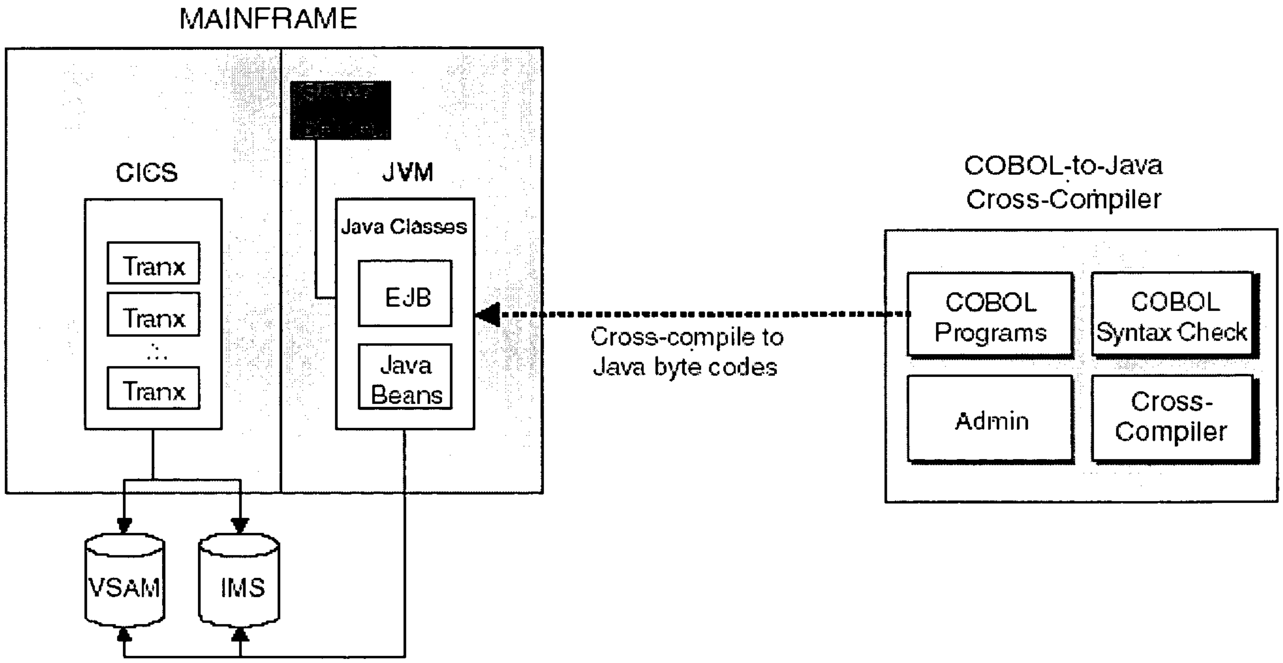

MAINFRAME
CICS
Tranx
Tranx
.:.
Tranx
JVM
Java Classes
EJB
Java Beans
VSAM
IMS
Cross-compile to Java byte codes
COBOL-to-Java Cross-Compiler
COBOL Programs
COBOL Syntax Check
Admin
Cross-Compiler

FIG. 56

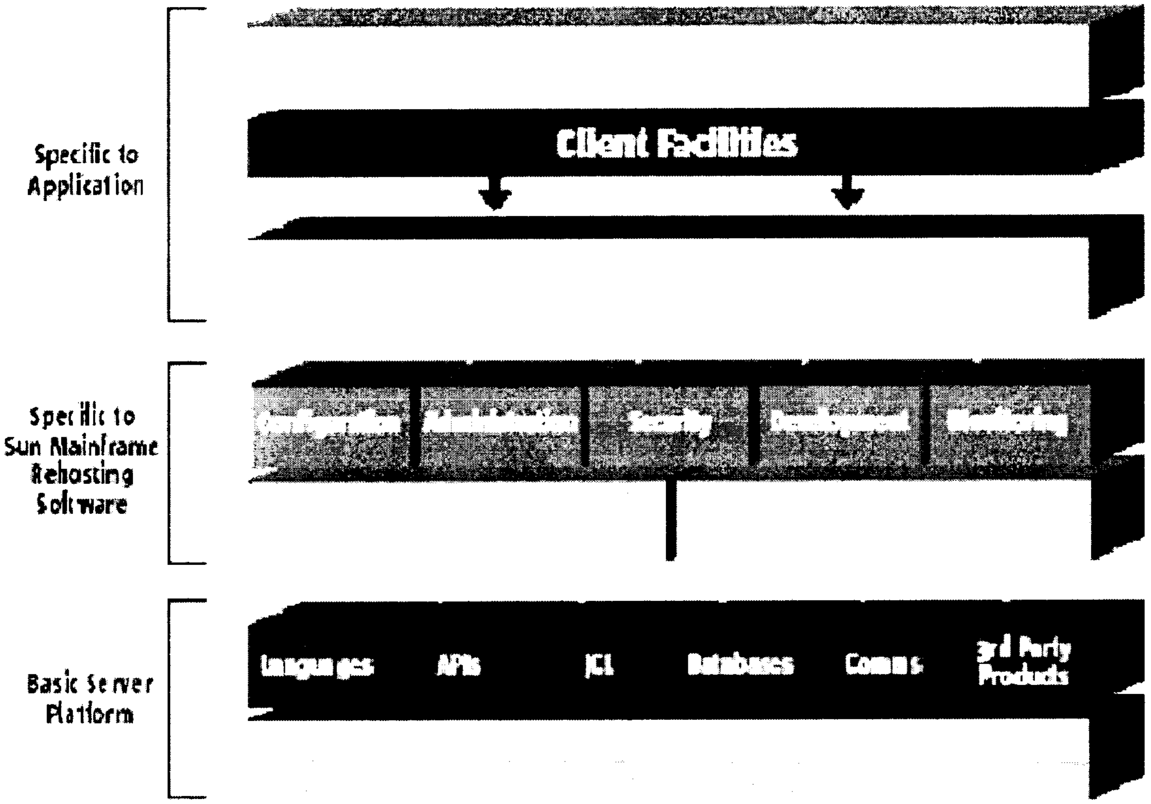

Specific to Application
Client Facilities
Specific to Sun Mainframe Rehosting Software
Configuration
Administration
Security
Development
Monitoring
Basic Server Platform
Languages
APIs
JCL
Databases
Comms
3rd Party Products

FIG. 57

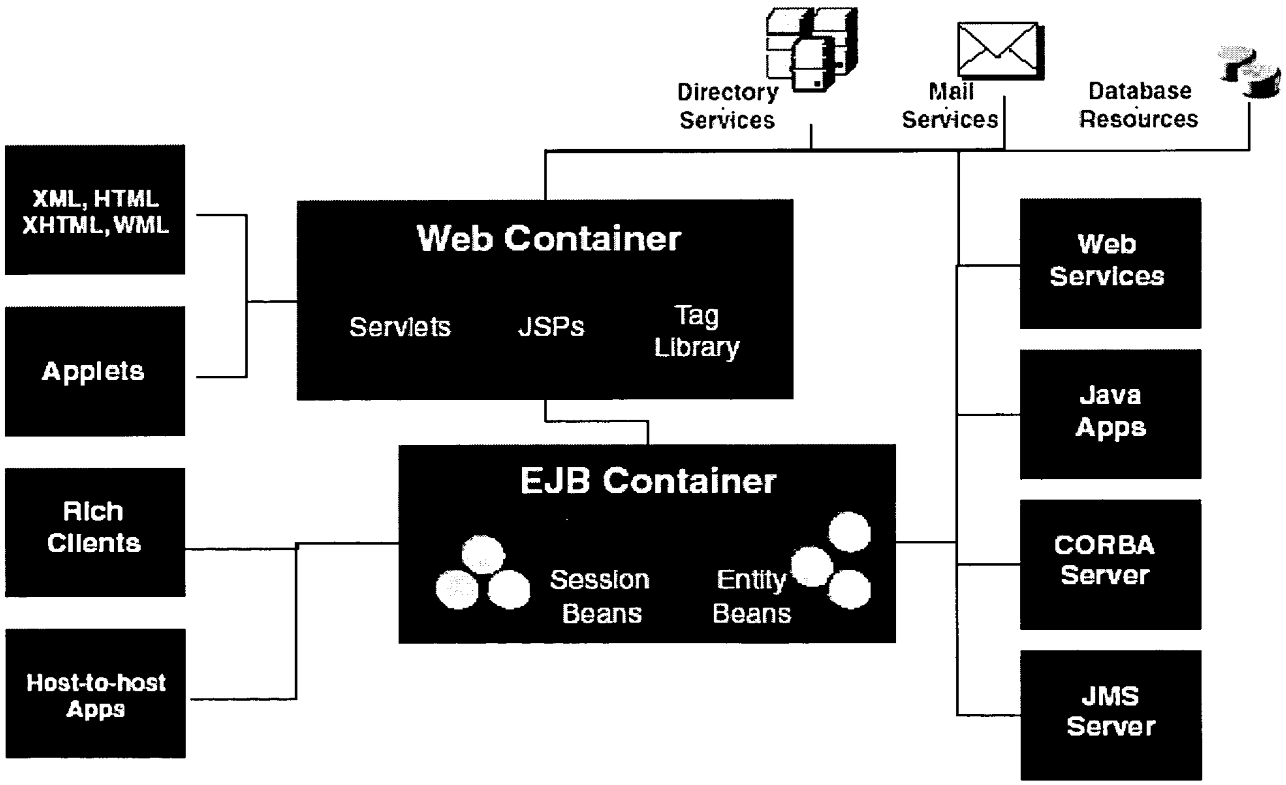
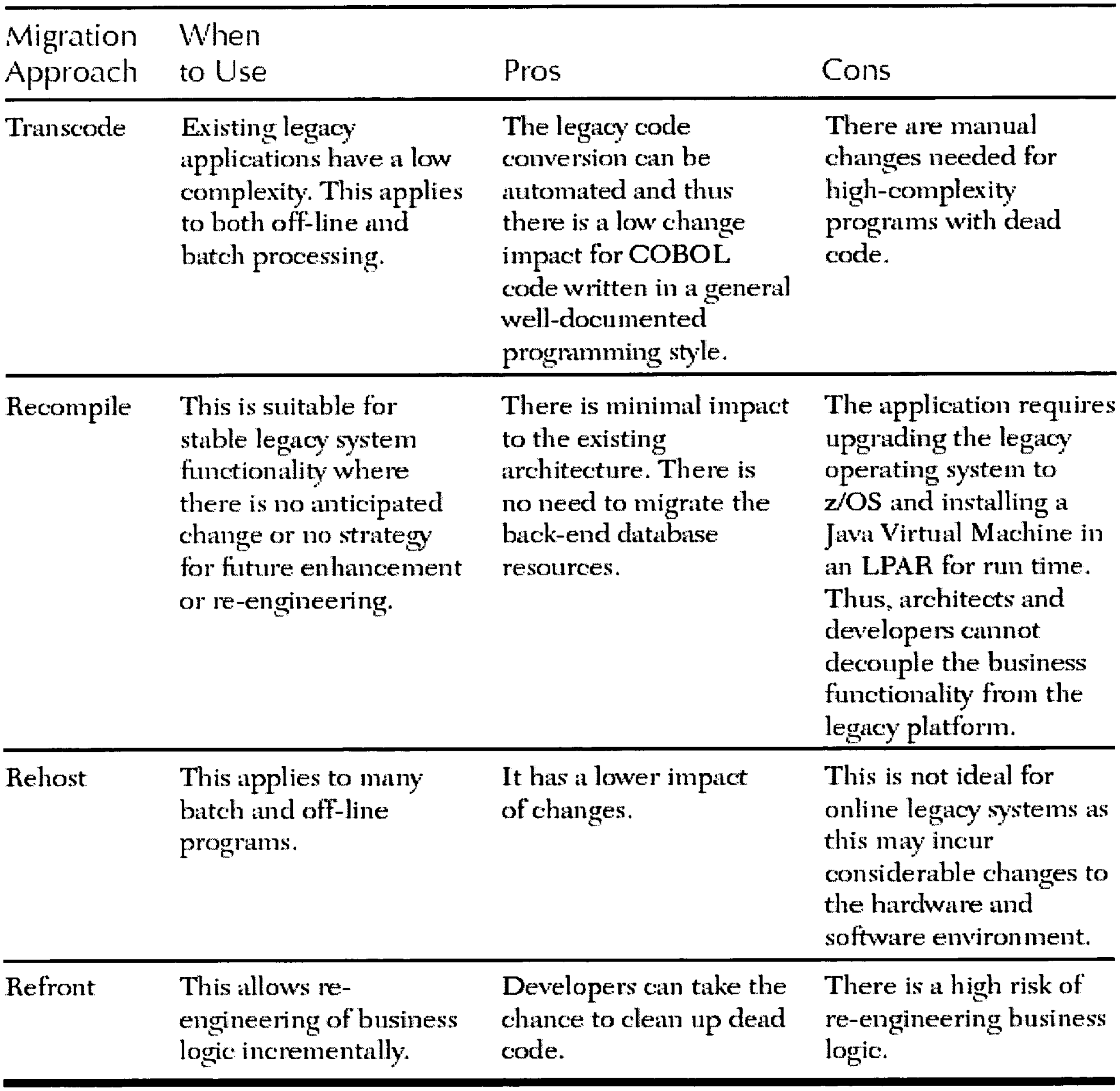
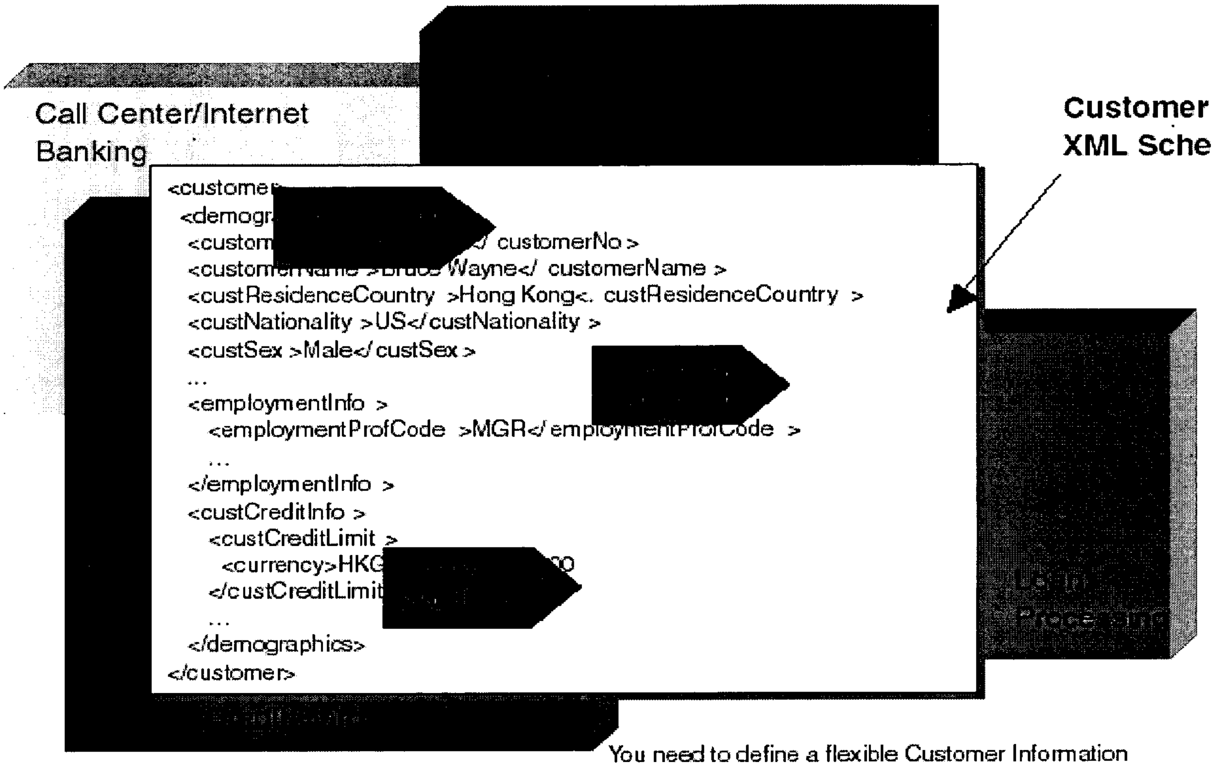

Directory Services
Mail Services
Database Resources
XML, HTML XHTML, WML
Applets
Rich Clients
Host-to-host Apps
Web Container
Servlets
JSPs
Tag Library
EJB Container
Session Beans
Entity Beans
Web Services
Java Apps
CORBA Server
JMS Server

FIG. 58



Migration Approach	When to Use	Pros	Cons
Transcode	Existing legacy applications have a low complexity. This applies to both off-line and batch processing.	The legacy code conversion can be automated and thus there is a low change impact for COBOL code written in a general well-documented programming style.	There are manual changes needed for high-complexity programs with dead code.
Recompile	This is suitable for stable legacy system functionality where there is no anticipated change or no strategy for future enhancement or re-engineering.	There is minimal impact to the existing architecture. There is no need to migrate the back-end database resources.	The application requires upgrading the legacy operating system to z/OS and installing a Java Virtual Machine in an LPAR for run time. Thus, architects and developers cannot decouple the business functionality from the legacy platform.
Rehost	This applies to many batch and off-line programs.	It has a lower impact of changes.	This is not ideal for online legacy systems as this may incur considerable changes to the hardware and software environment.
Refront	This allows re-engineering of business logic incrementally.	Developers can take the chance to clean up dead code.	There is a high risk of re-engineering business logic.

FIG. 59



```
<custome
 <demogr
  <custom                   / customerNo >
  <custom                   Wayne</ customerName >
  <custResidenceCountry >Hong Kong<. custResidenceCountry >
  <custNationality >US</custNationality >
  <custSex >Male</ custSex >
  ...
  <employmentInfo >
    <employmentProfCode  >MGR</ emp                Code  >
    ...
  </employmentInfo >
  <custCreditInfo >
    <custCreditLimit >
      <currency>HKG
    </custCreditLimit
    ...
  </demographics>
</customer>
```

You need to define a flexible Customer Information XML schema, which can be sharable by different Banking services.

FIG. 60

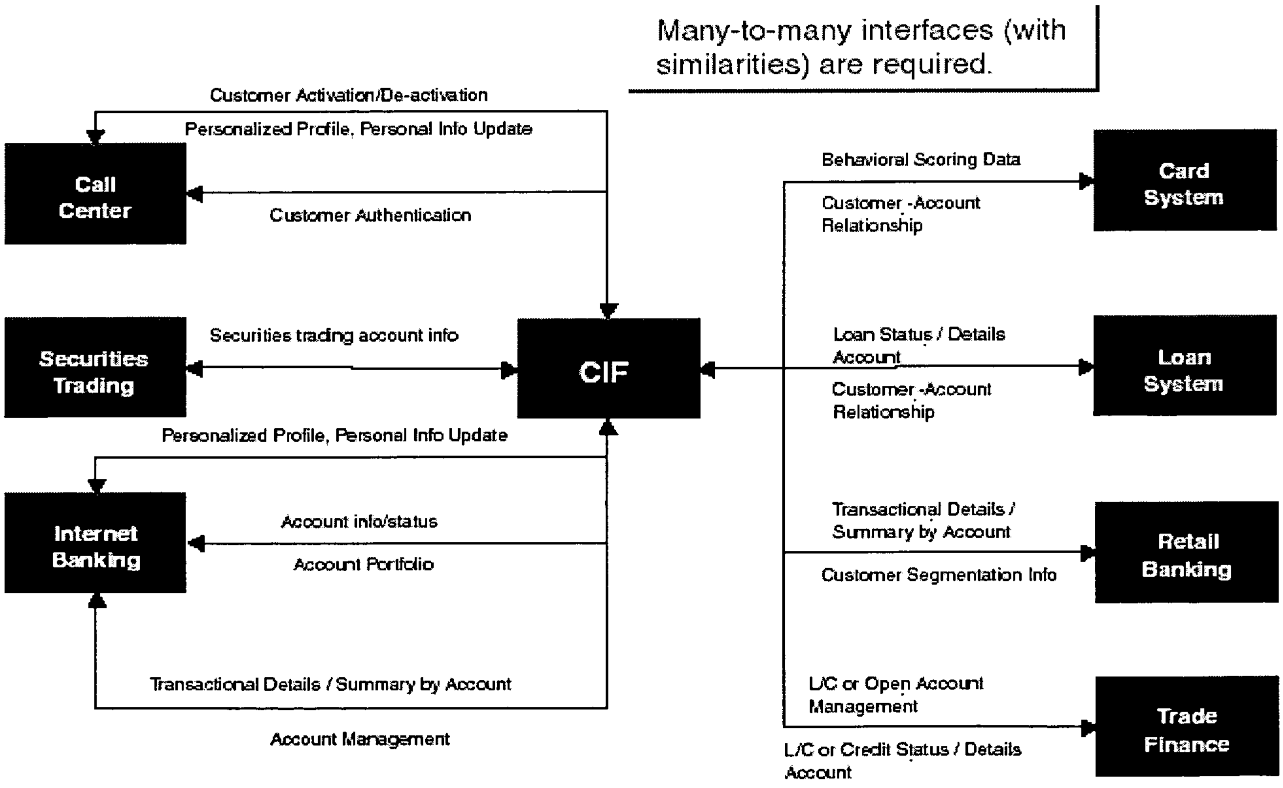

Many-to-many interfaces (with similarities) are required.
Customer Activation/De-activation
Personalized Profile, Personal Info Update
Call Center
Customer Authentication
Securities Trading
Securities trading account info
CIF
Personalized Profile, Personal Info Update
Internet Banking
Account info/status
Account Portfolio
Transactional Details / Summary by Account
Account Management
Behavioral Scoring Data
Customer -Account Relationship
Card System
Loan Status / Details Account
Customer -Account Relationship
Loan System
Transactional Details / Summary by Account
Customer Segmentation Info
Retail Banking
L/C or Open Account Management
L/C or Credit Status / Details Account
Trade Finance

FIG. 61

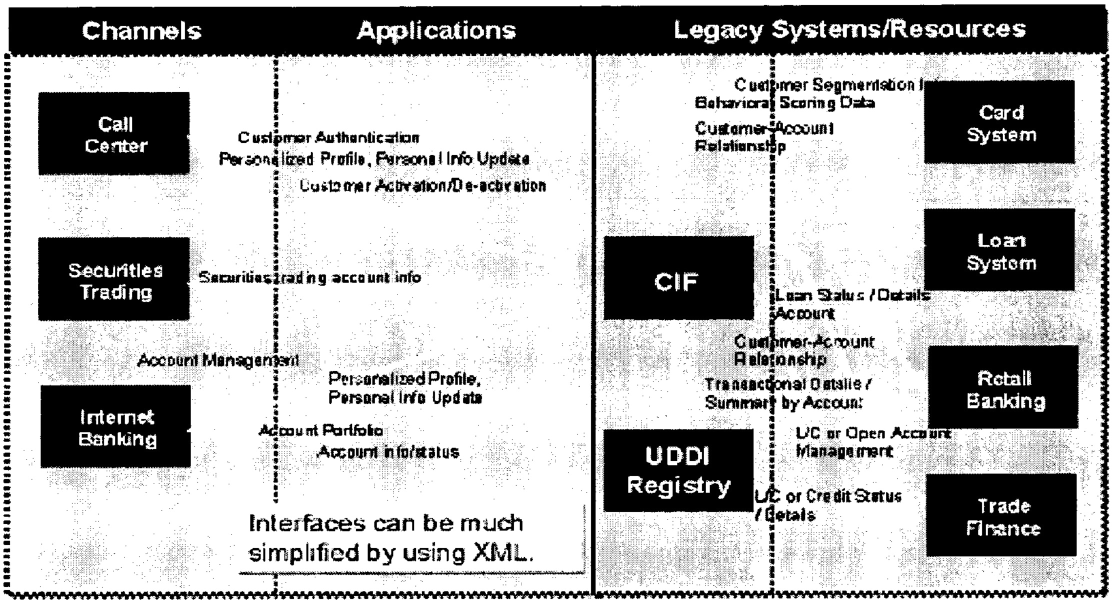

Channels
Applications
Legacy Systems/Resources
Call Center
Securities Trading
Internet Banking
Customer Authentication
Personalized Profile, Personal Info Update
Customer Activation/De-activation
Securities trading account info
Account Management
Personalized Profile, Personal Info Update
Account Portfolio
Account info/status
Interfaces can be much simplified by using XML.
Customer Segmentation I
Behavioral Scoring Data
Customer-Account Relationship
Card System
Loan System
CIF
Loan Status / Details Account
Customer-Account Relationship
Transactional Details / Summary by Account
Retail Banking
L/C or Open Account Management
UDDI Registry
L/C or Credit Status / Details
Trade Finance

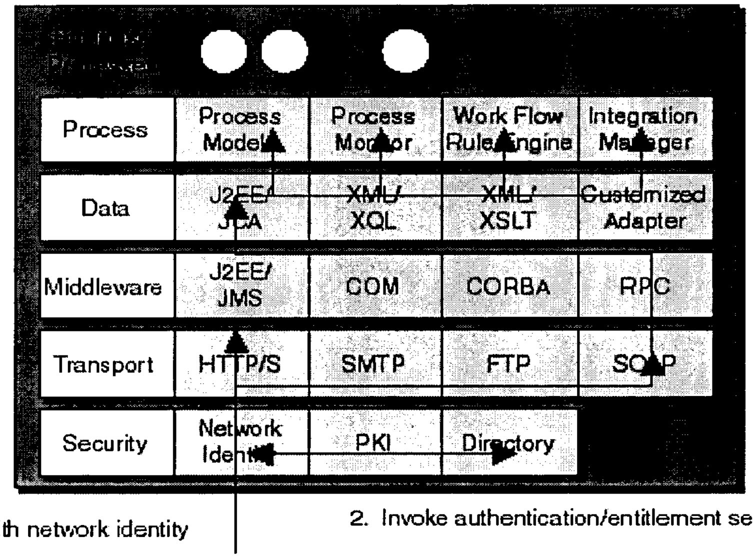
FIG. 62

Sample Scenario–Fund Transfer

6. Complicate back-end business processes are managed by a workflow rule engine and an integration manager

5. JCA connector connects Web Services client requests to back-end legacy systems

3. Client request in SOAP messages carried over HTTP/S



Process	Process Model	Process Monitor	Work Flow Rule Engine	Integration Manager
Data	J2EE/ JCA	XML/ XQL	XML/ XSLT	Customized Adapter
Middleware	J2EE/ JMS	COM	CORBA	RPC
Transport	HTTP/S	SMTP	FTP	SOAP
Security	Network Identity	PKI	Directory	

4. Server-side SOAP component invokes legacy system functionality via XML-RPC

1. Single Sign-on with network identity

2. Invoke authentication/entitlement services

FIG. 63

	Client Tier	Presentation Tier	Business Tier	Integration Tier	Resources Tier
Application Layer				Process Models Process Monitor Workflow Rule Engine Integration Manager	
Virtual Layer		XSLT	XML	JMS RPC COM CORBA	JCA XQL
Upper Layer	HTTPS SOAP	HTTPS SOAP SMTP FTP	SOAP		
Lower Layer	Network Identity/ Single Sign-on PKI Directory server	Network Identity/ Single Sign-on PKI Directory server	Network Identity/ Single Sign-on PKI Directory server	Network Identity/ Single Sign-on PKI Directory server	Network Identity/ Single Sign-on PKI Directory server

FIG. 64

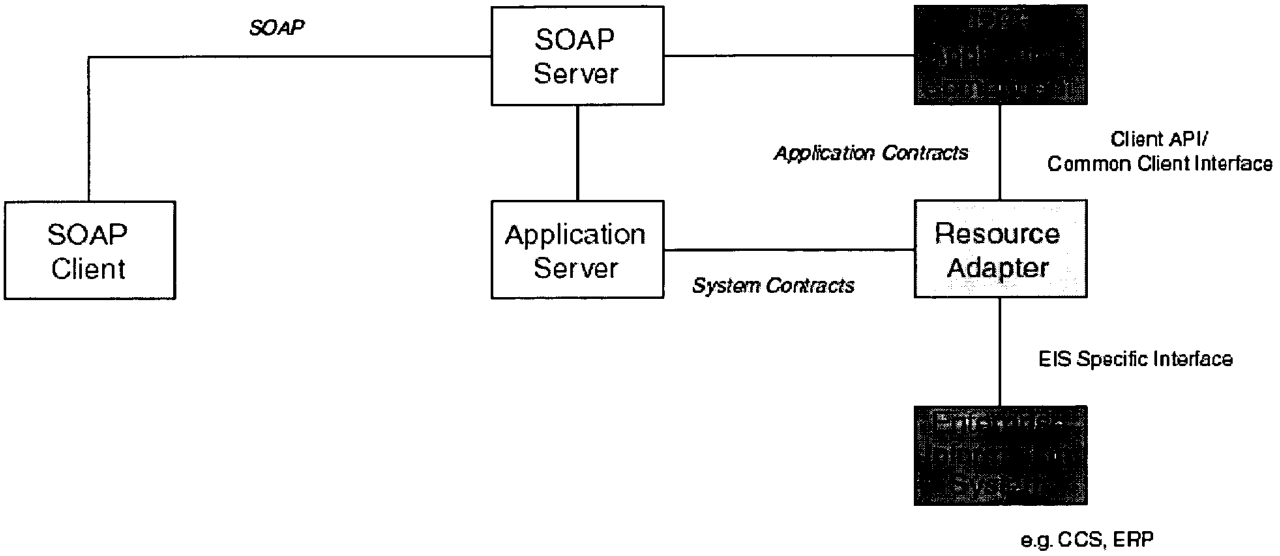

SOAP
SOAP Server
J2EE Application Component
Application Contracts
Client API/ Common Client Interface
SOAP Client
Application Server
Resource Adapter
System Contracts
EIS Specific Interface
Enterprise Information System
e.g. CCS, ERP

FIG. 65

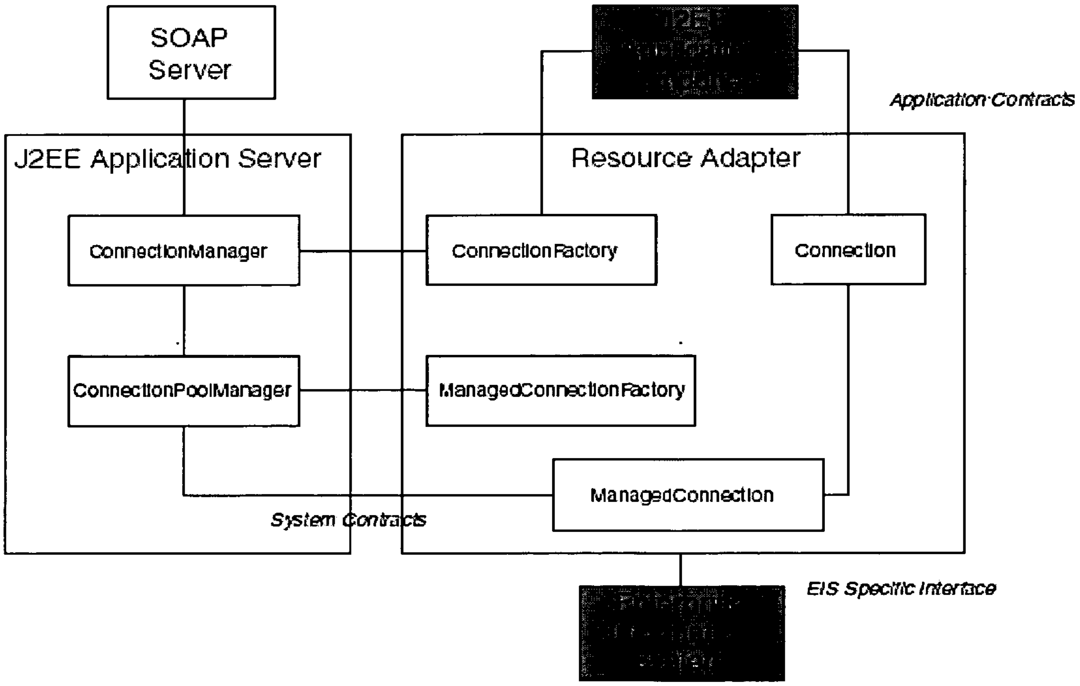
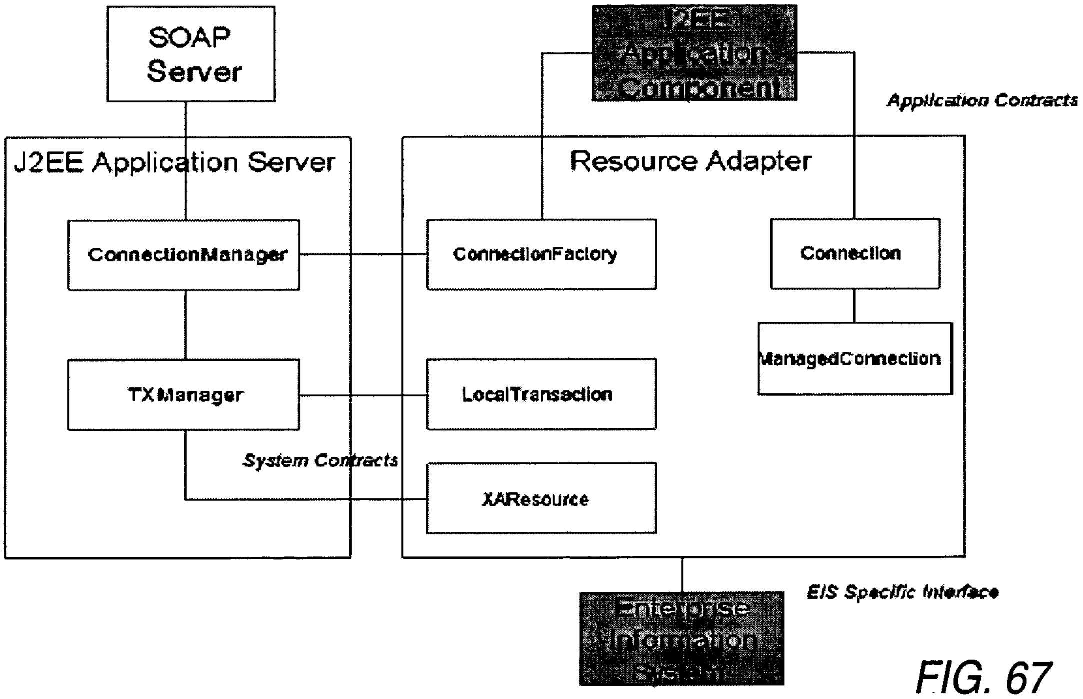
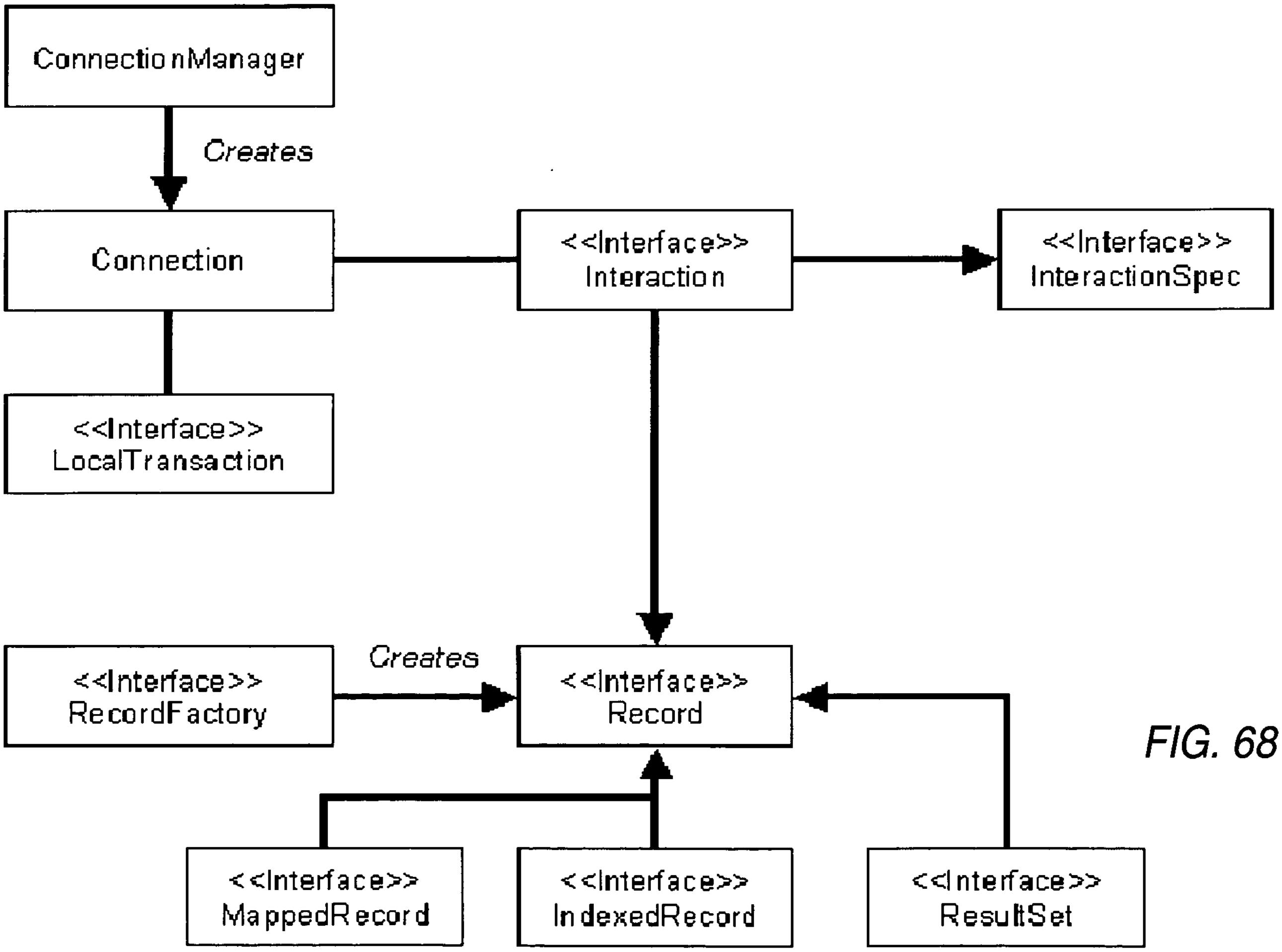

SOAP Server
J2EE Application Component
Application Contracts
J2EE Application Server
Resource Adapter
ConnectionManager
ConnectionFactory
Connection
ConnectionPoolManager
ManagedConnectionFactory
ManagedConnection
System Contracts
EIS Specific Interface
Enterprise Information System

FIG. 66



SOAP Server
J2EE Application Component
Application Contracts
J2EE Application Server
Resource Adapter
ConnectionManager
ConnectionFactory
Connection
ManagedConnection
TXManager
LocalTransaction
System Contracts
XAResource
EIS Specific Interface
Enterprise Information System

FIG. 67



ConnectionManager
Creates
Connection
<<Interface>> Interaction
<<Interface>> InteractionSpec
<<Interface>> LocalTransaction
<<Interface>> RecordFactory
Creates
<<Interface>> Record
<<Interface>> MappedRecord
<<Interface>> IndexedRecord
<<Interface>> ResultSet

FIG. 68

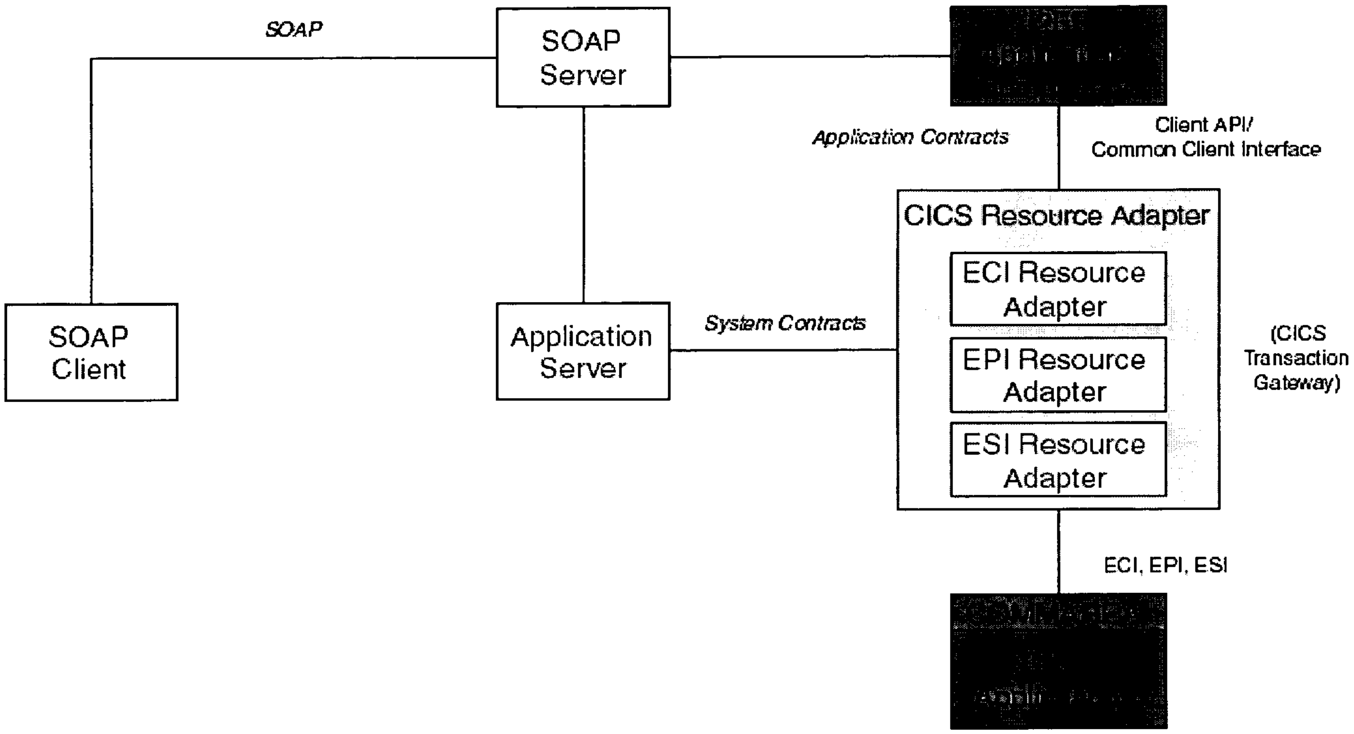

SOAP
SOAP Server
Application Contracts
Client API/ Common Client Interface
CICS Resource Adapter
ECI Resource Adapter
EPI Resource Adapter
ESI Resource Adapter
(CICS Transaction Gateway)
SOAP Client
Application Server
System Contracts
ECI, EPI, ESI

FIG. 69

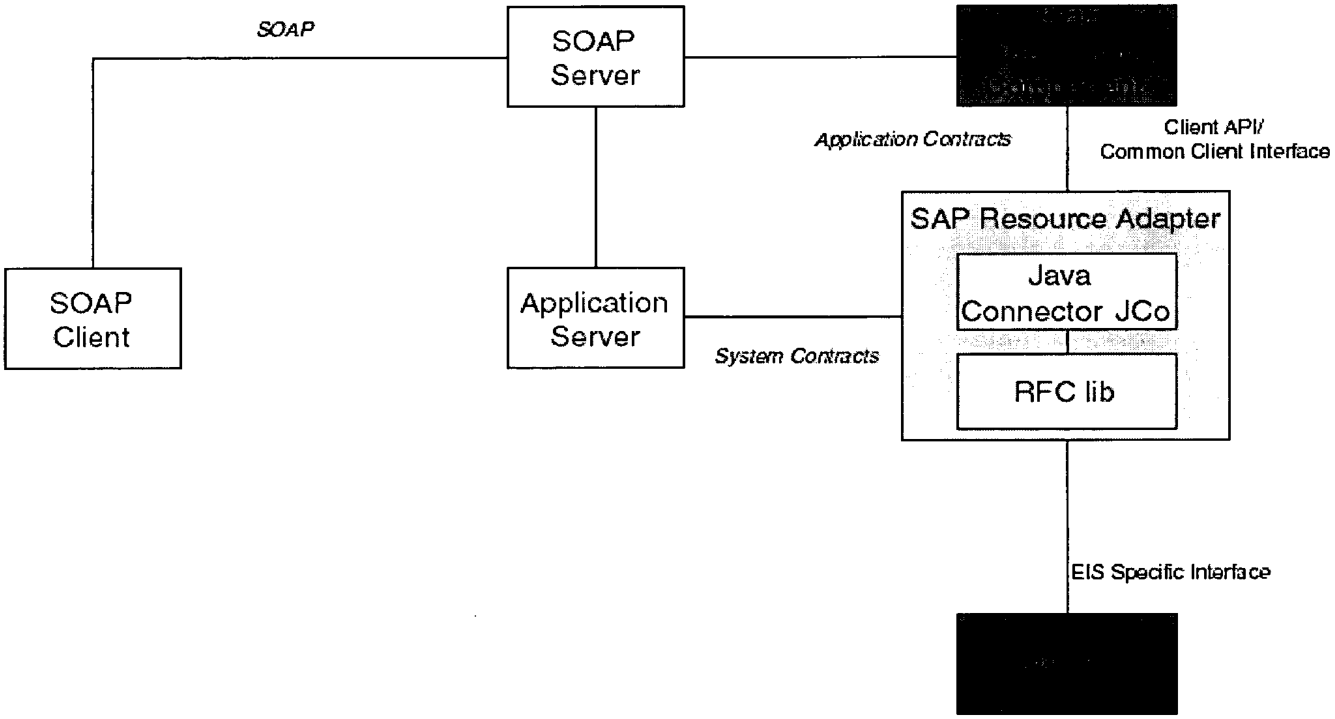
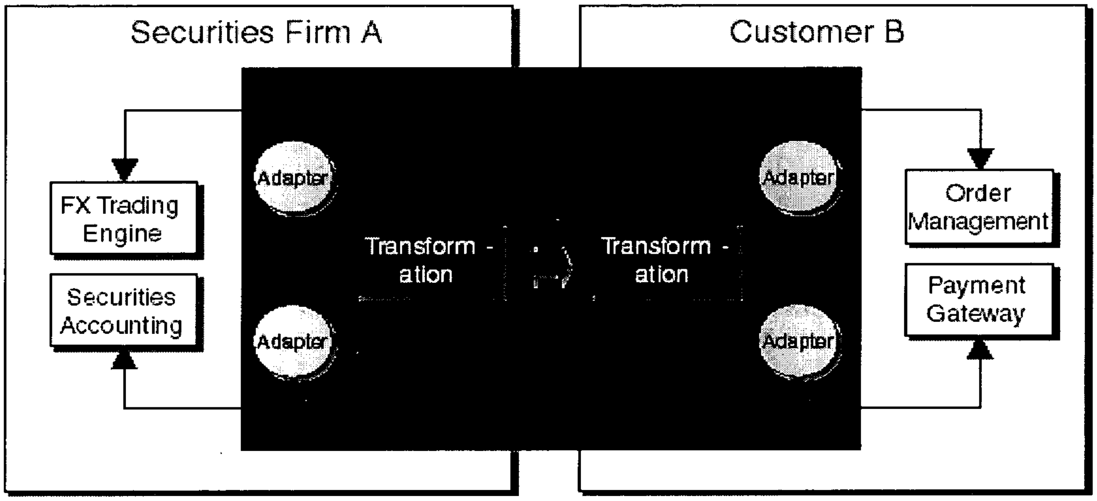

SOAP
SOAP Server
Application Contracts
Client API/ Common Client Interface
SAP Resource Adapter
Java Connector JCo
RFC lib
SOAP Client
Application Server
System Contracts
EIS Specific Interface

FIG. 70



Typical Technology Used
Customized adapter
Preagreed interface format/standard
EDI translator

When to Use
Point-to-point exchange, tight integration
Limited number of trading partners
Relatively static data formats

Based on: Yee & Apte. Integrating Your e-Business Enterprise. SAMS, 2001.

FIG. 71

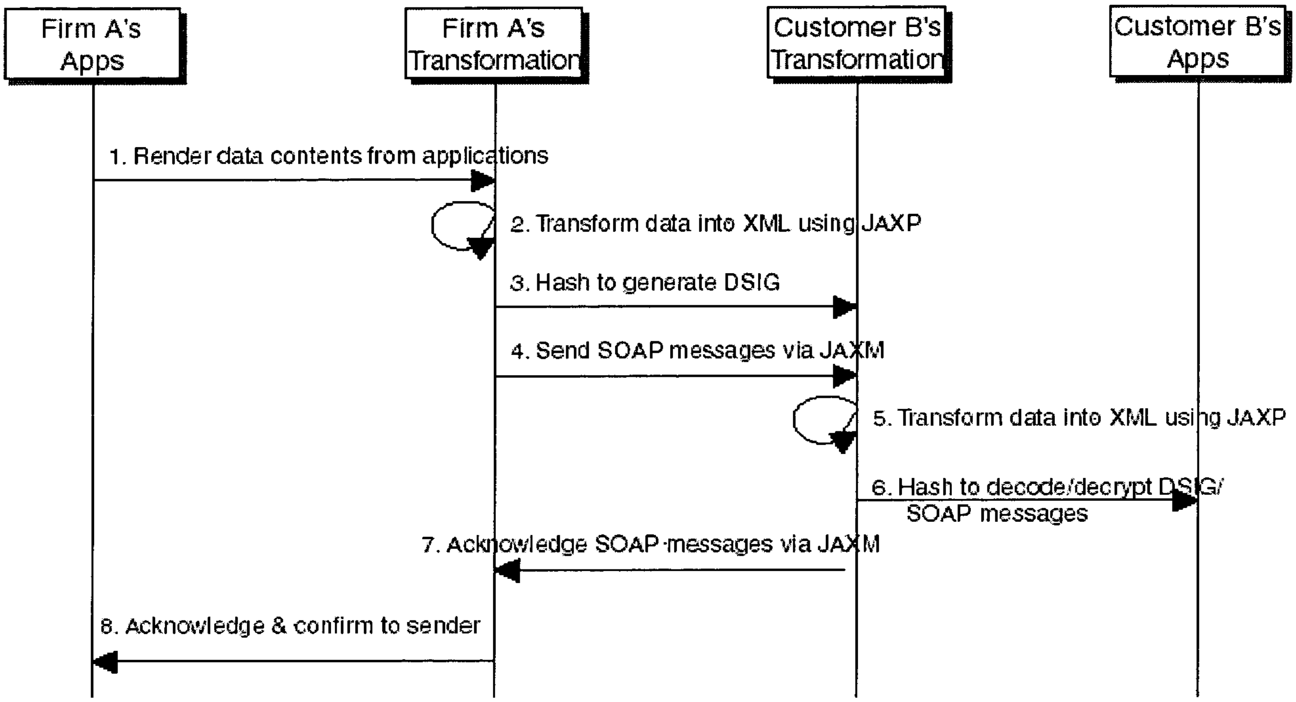
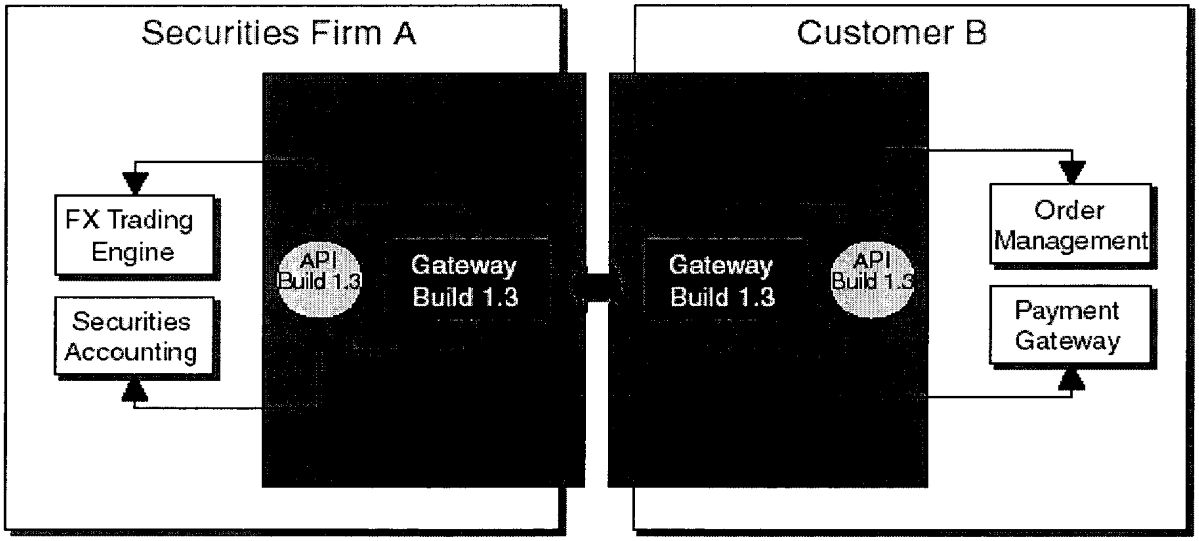

Firm A's Apps
Firm A's Transformation
Customer B's Transformation
Customer B's Apps
1. Render data contents from applications
2. Transform data into XML using JAXP
3. Hash to generate DSIG
4. Send SOAP messages via JAXM
5. Transform data into XML using JAXP
6. Hash to decode/decrypt DSIG/ SOAP messages
7. Acknowledge SOAP messages via JAXM
8. Acknowledge & confirm to sender

FIG. 72



Typical Technology Used
Standardized home-grown/customized adapter
Standardized interface format/API standard
EDI translator/EAI or middleware

When to Use
Strong urge for standard build
Point-to-point exchange, tight integration

FIG. 73

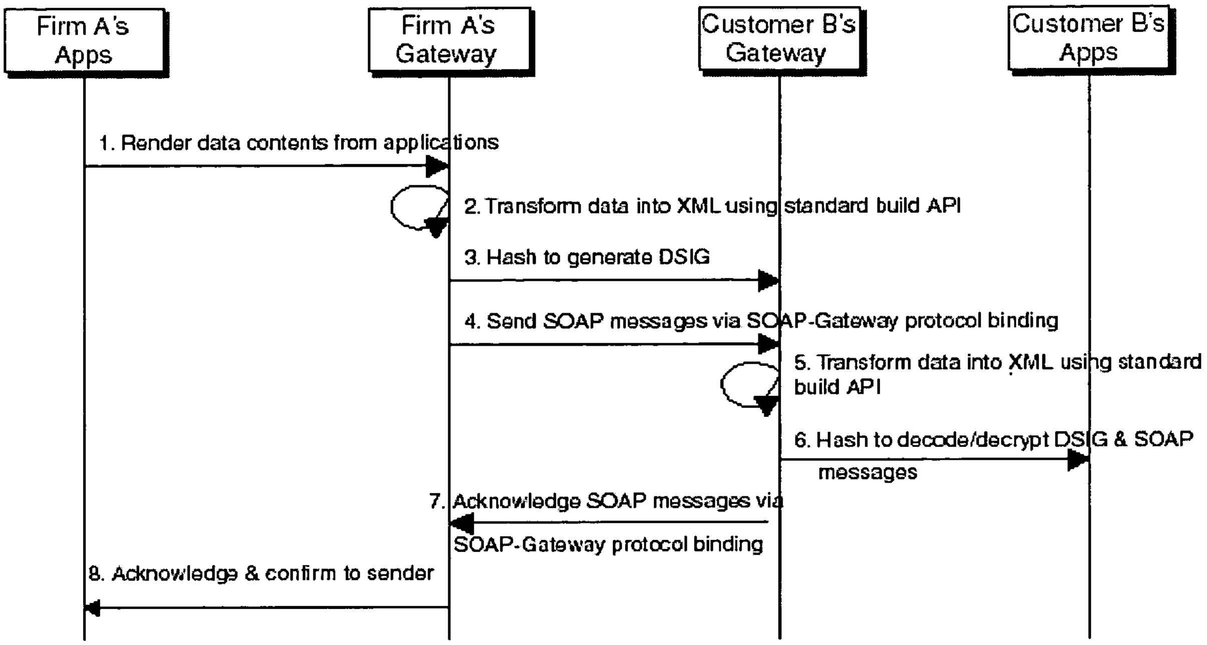
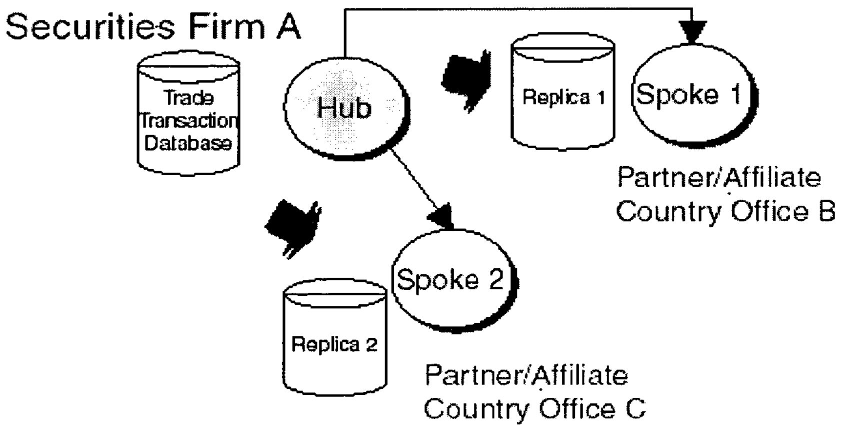

Firm A's Apps
Firm A's Gateway
Customer B's Gateway
Customer B's Apps
1. Render data contents from applications
2. Transform data into XML using standard build API
3. Hash to generate DSIG
4. Send SOAP messages via SOAP-Gateway protocol binding
5. Transform data into XML using standard build API
6. Hash to decode/decrypt DSIG & SOAP messages
7. Acknowledge SOAP messages via SOAP-Gateway protocol binding
8. Acknowledge & confirm to sender

FIG. 74


Securities Firm A
Trade Transaction Database
Hub
Replica 1
Spoke 1
Partner/Affiliate Country Office B
Spoke 2
Replica 2
Partner/Affiliate Country Office C

Typical Technology Used
Synchronous/asynchronous database replication (push-pull)
Database/message centric applications
EAI/Messaging middleware (e.g., RV-TX JMS with JMS Bridge or JMS-SOAP)

When to Use
Highly centralized business applications
No geographical location constraints
Local spokes are for backup/ performance benefits (e.g., faster access, MIS)

FIG. 75

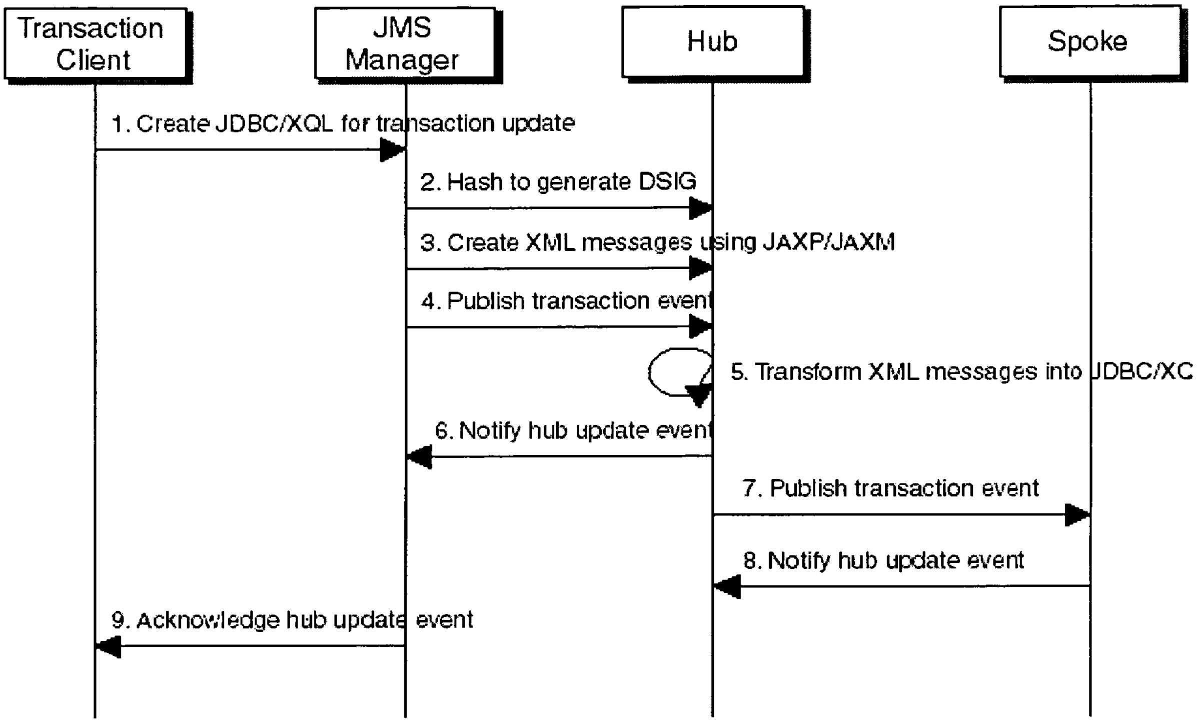
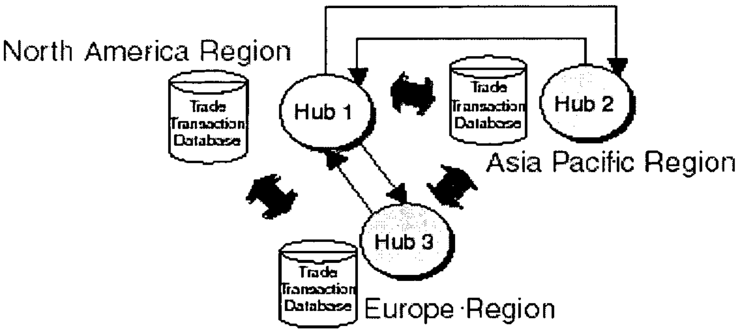

Transaction Client
JMS Manager
Hub
Spoke
1. Create JDBC/XQL for transaction update
2. Hash to generate DSIG
3. Create XML messages using JAXP/JAXM
4. Publish transaction event
5. Transform XML messages into JDBC/XC
6. Notify hub update event
7. Publish transaction event
8. Notify hub update event
9. Acknowledge hub update event

FIG. 76


North America Region
Trade Transaction Database
Hub 1
Trade Transaction Database
Hub 2
Asia Pacific Region
Hub 3
Trade Transaction Database
Europe Region

Typical Technology Used
Synchronous/asynchronous database replication (push-push)
Database/Message centric applications
EAI/Messaging middleware, (e.g., RV-TX JMS with JMS Bridge or JMS-SOAP)

When to Use
Highly distributed business applications with local control
Geographical location constraints
Partition different hubs for different products or transaction types, where replications are for back-up purpose

FIG. 77

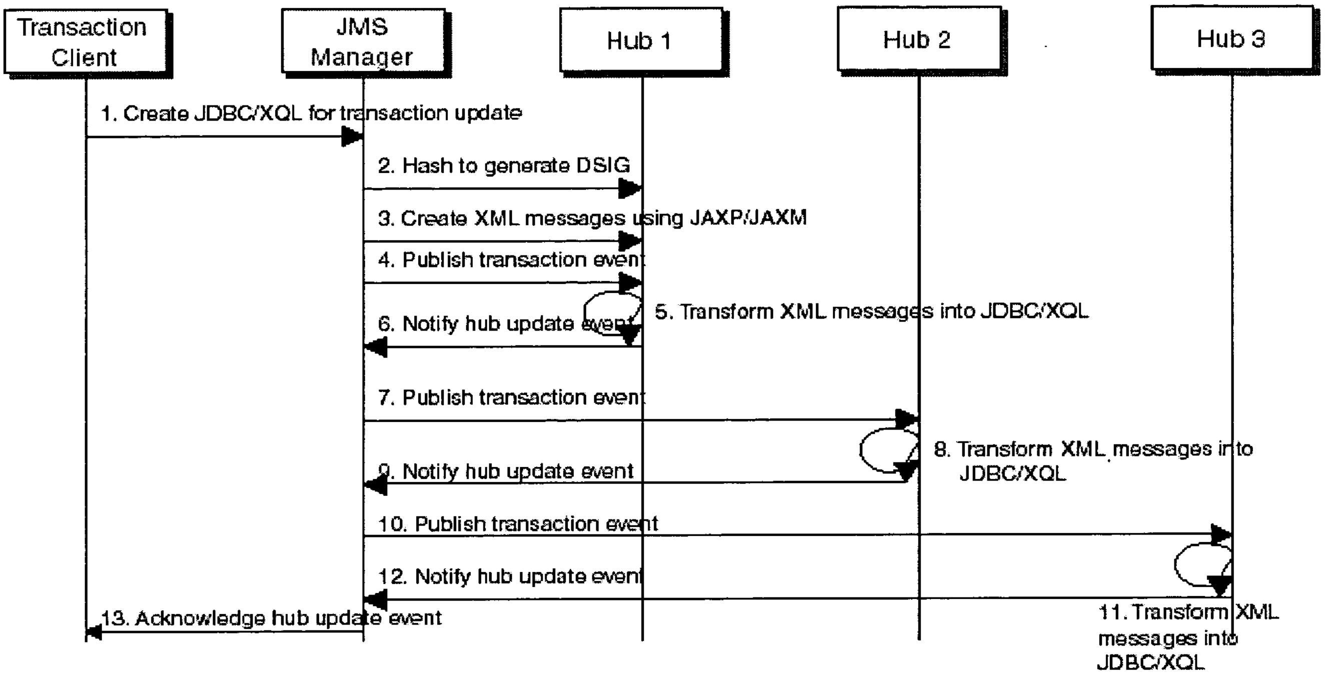
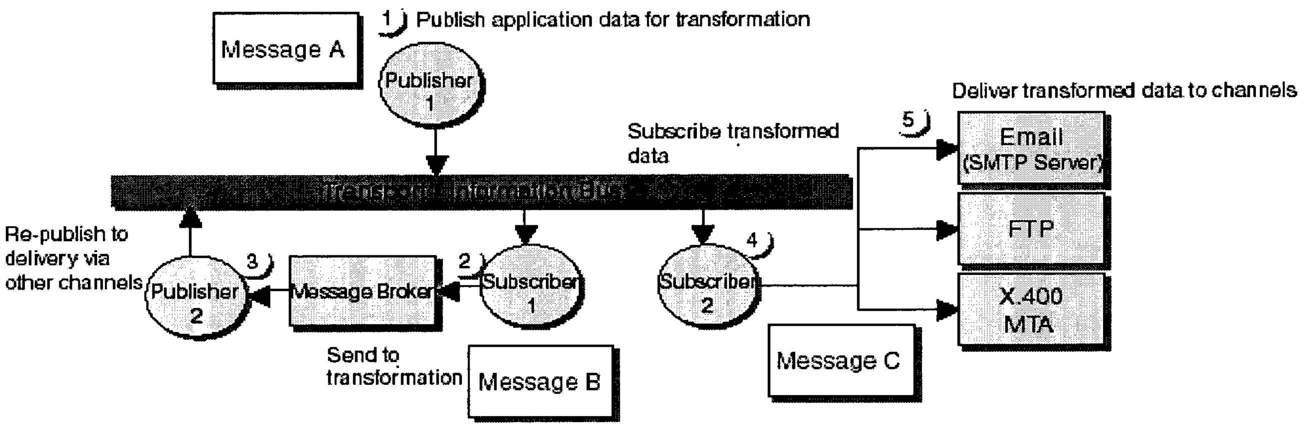

Transaction Client
JMS Manager
Hub 1
Hub 2
Hub 3
1. Create JDBC/XQL for transaction update
2. Hash to generate DSIG
3. Create XML messages using JAXP/JAXM
4. Publish transaction event
5. Transform XML messages into JDBC/XQL
6. Notify hub update event
7. Publish transaction event
8. Transform XML messages into JDBC/XQL
9. Notify hub update event
10. Publish transaction event
12. Notify hub update event
11. Transform XML messages into JDBC/XQL
13. Acknowledge hub update event

FIG. 78


Message A
1 Publish application data for transformation
Publisher 1
Deliver transformed data to channels
Subscribe transformed data
5
Email (SMTP Server)
Transform Information Bus
FTP
Re-publish to delivery via other channels
3
2
4
Publisher 2
Message Broker
Subscriber 1
Subscriber 2
X.400 MTA
Send to transformation
Message B
Message C

Typical Technology Used

EAI/Messaging middleware, e.g., Amtrix, Mercator

EDI Translator

JMS or non-JMS middleware

When to Use

Complicate data transformation or work flow

Multi-channel delivery support, (e.g., email, fax, EDI)

FIG. 79

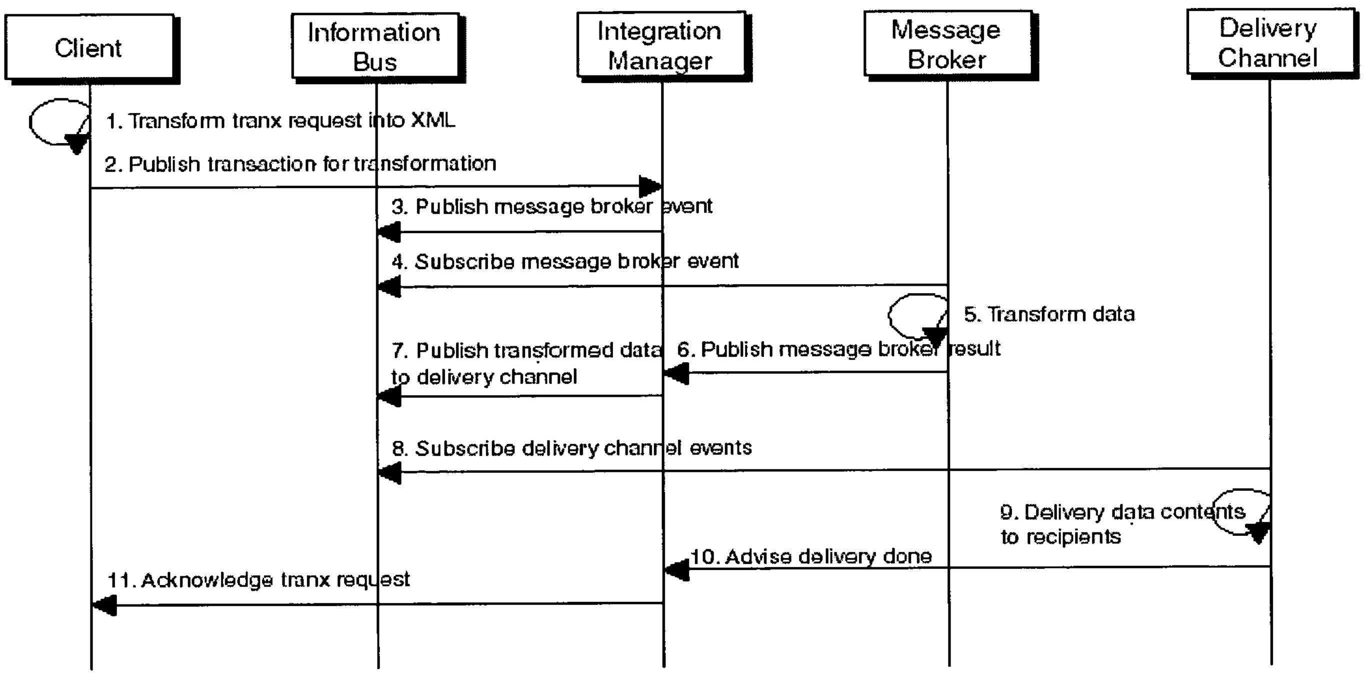
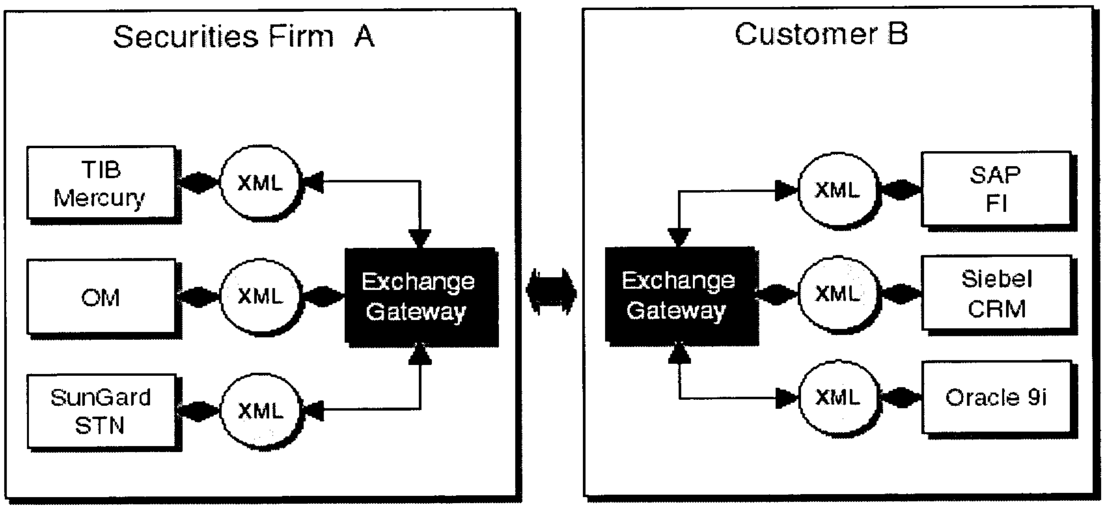

Client
Information Bus
Integration Manager
Message Broker
Delivery Channel
1. Transform tranx request into XML
2. Publish transaction for transformation
3. Publish message broker event
4. Subscribe message broker event
5. Transform data
6. Publish message broker result
7. Publish transformed data to delivery channel
8. Subscribe delivery channel events
9. Delivery data contents to recipients
10. Advise delivery done
11. Acknowledge tranx request

FIG. 80



Typical Technology Used
Vendor/off-the-shelf XML adapter
Preagreed XML standards/variants
XML Web Services

When to Use
Loosely coupled integration
Large number of trading partners
Multiple systems need to be integrated

Based on: Yee & Apte. Integrating Your e-Business Enterprise. SAMS, 2001.

FIG. 81

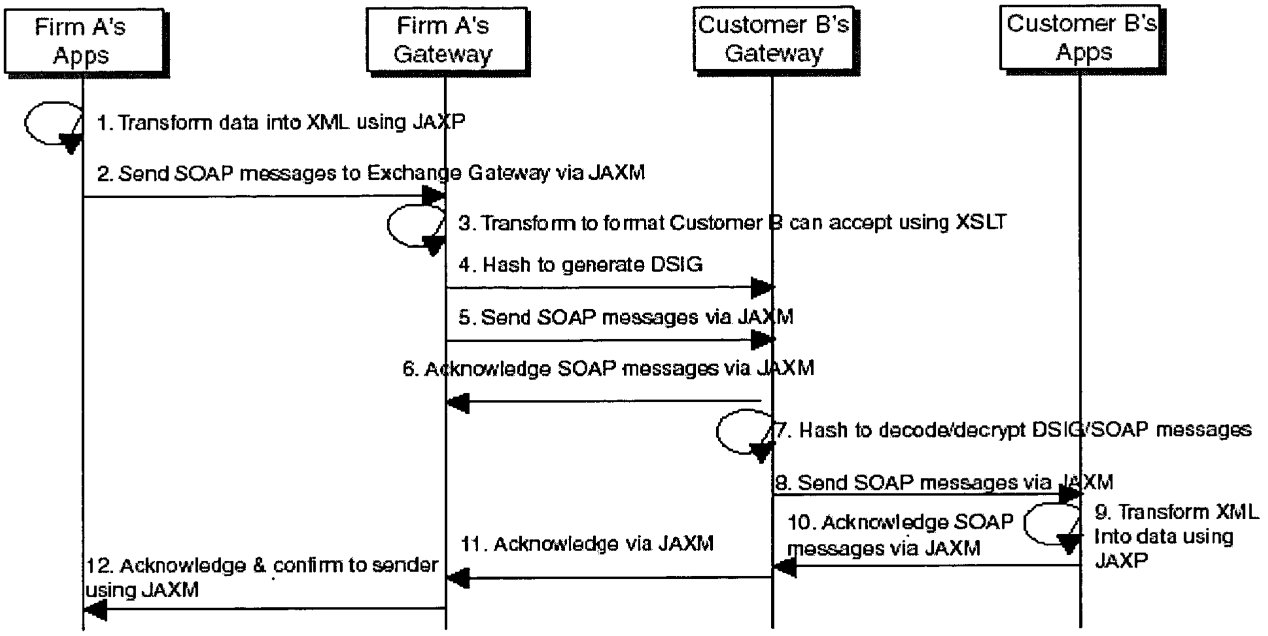
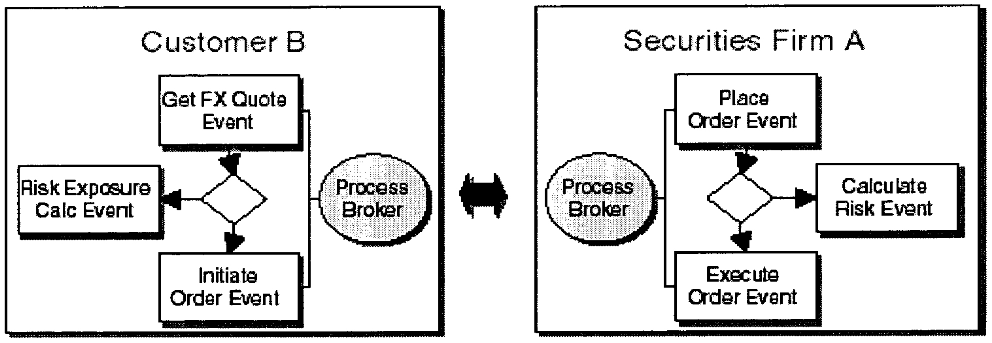

Firm A's Apps
Firm A's Gateway
Customer B's Gateway
Customer B's Apps
1. Transform data into XML using JAXP
2. Send SOAP messages to Exchange Gateway via JAXM
3. Transform to format Customer B can accept using XSLT
4. Hash to generate DSIG
5. Send SOAP messages via JAXM
6. Acknowledge SOAP messages via JAXM
7. Hash to decode/decrypt DSIG/SOAP messages
8. Send SOAP messages via JAXM
9. Transform XML Into data using JAXP
10. Acknowledge SOAP messages via JAXM
11. Acknowledge via JAXM
12. Acknowledge & confirm to sender using JAXM

FIG. 82



Typical Technology Used
Customized work flow integration tools
Preagreed message formats/APIs

When to Use
Tightly coupled integration
Small number of trading partners
Strong business service integration needs

Based on: Yee & Apte. Integrating Your e-Business Enterprise. SAMS, 2001.

FIG. 83

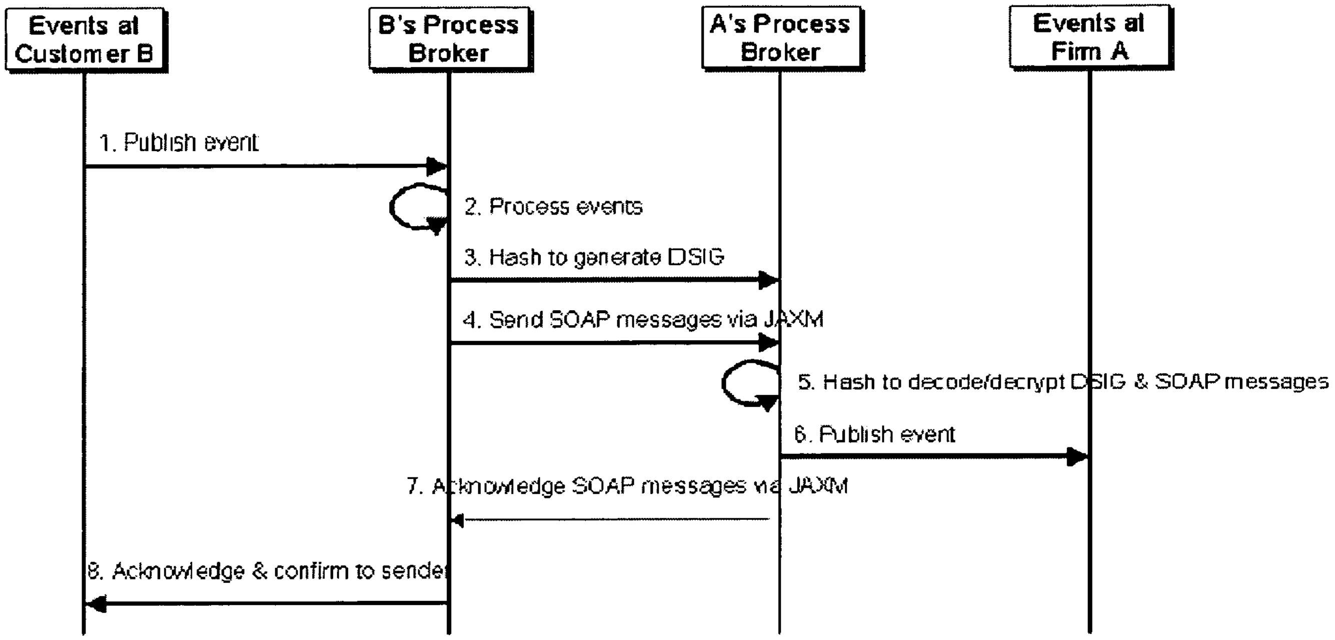
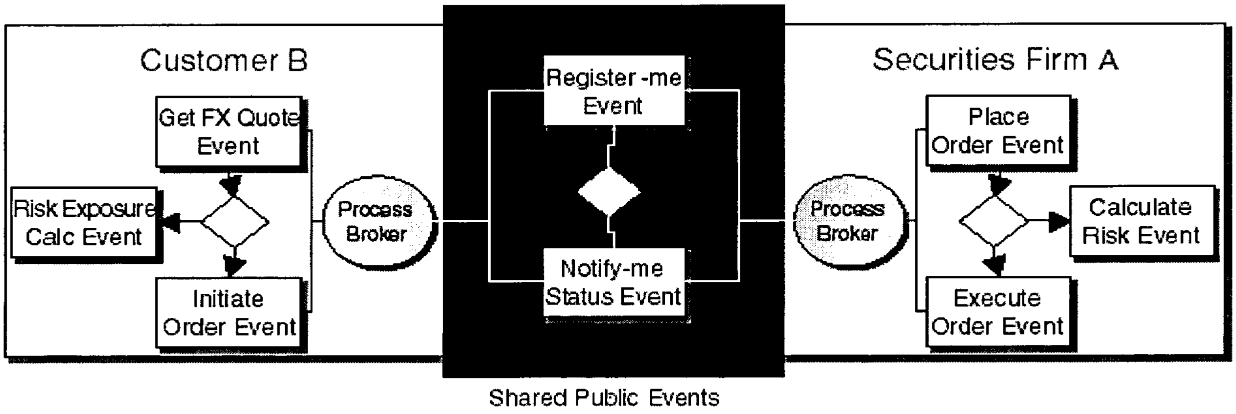

Events at Customer B
B's Process Broker
A's Process Broker
Events at Firm A
1. Publish event
2. Process events
3. Hash to generate DSIG
4. Send SOAP messages via JAXM
5. Hash to decode/decrypt DSIG & SOAP messages
6. Publish event
7. Acknowledge SOAP messages via JAXM
8. Acknowledge & confirm to sender

FIG. 84



Typical Technology Used
Customized workflow integration tools
Preagreed message formats/APIs
"Shared" process integration tools for public events

When to Use
"Co-branded" business services
Tightly coupled process & technical integration
Small number of trading partners

FIG. 85

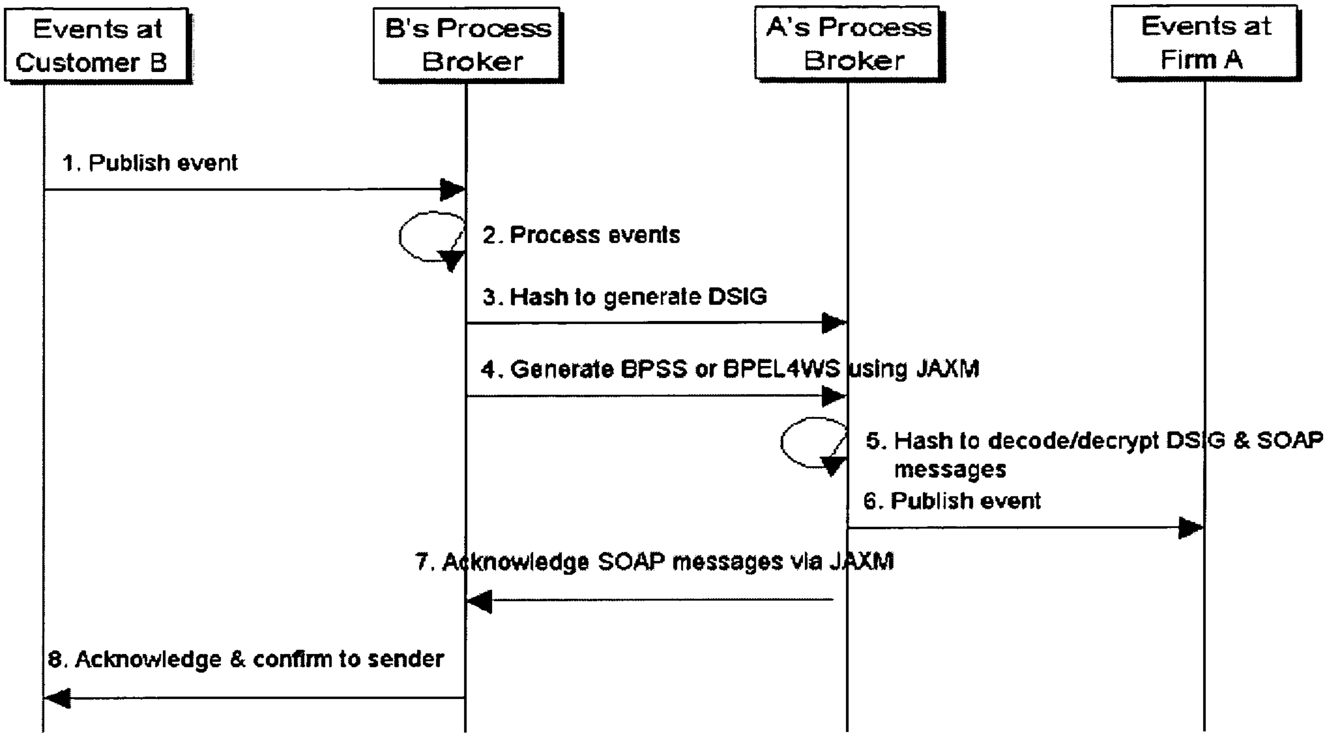
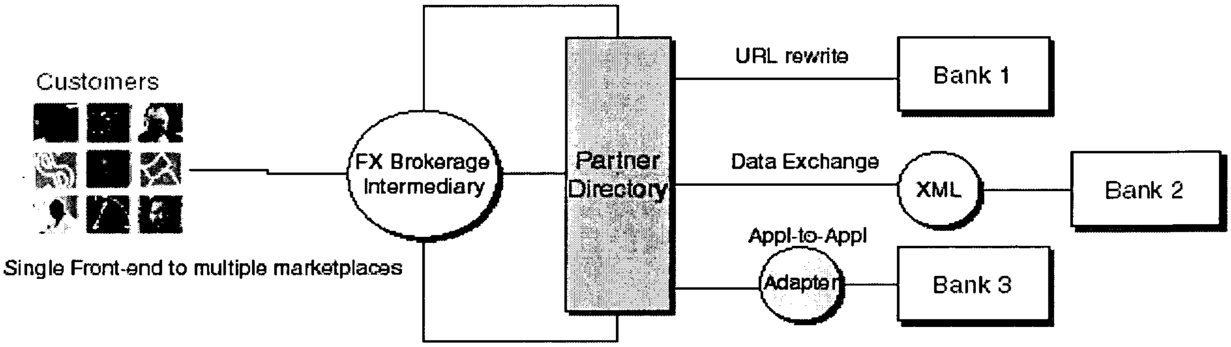

Events at Customer B
B's Process Broker
A's Process Broker
Events at Firm A
1. Publish event
2. Process events
3. Hash to generate DSIG
4. Generate BPSS or BPEL4WS using JAXM
5. Hash to decode/decrypt DSIG & SOAP messages
6. Publish event
7. Acknowledge SOAP messages via JAXM
8. Acknowledge & confirm to sender

FIG. 86



Typical Technology Used
Hyrbrid integration methods
Pre-agreed message formats/APIs
XML Web Services
HTTP/S GET or POST

When to Use
Brokering similar services with a single front-end
(service–provider neutral)
Loosely coupled process & technical integration
Large number of trading partners

FIG. 87

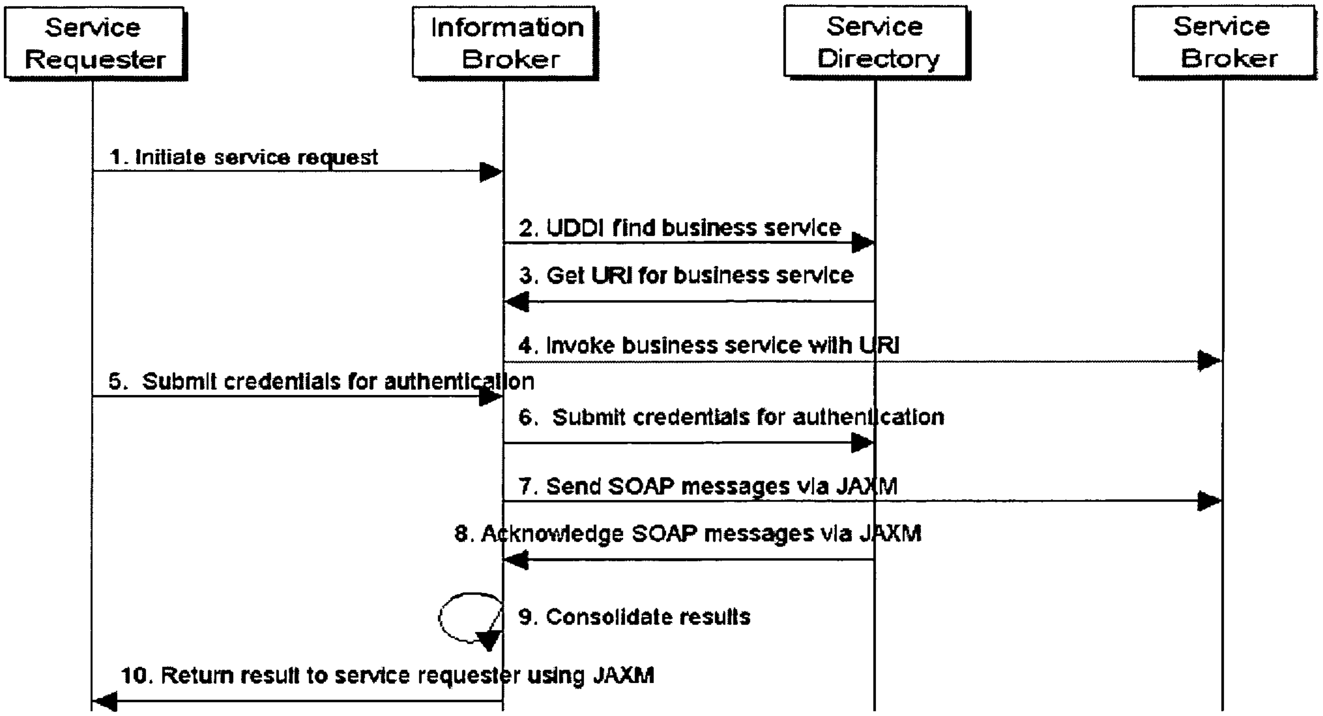
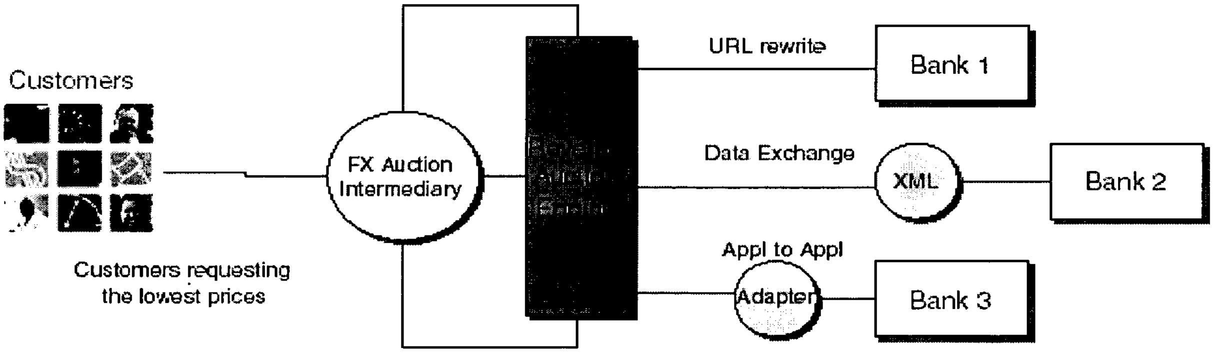

Service Requester
Information Broker
Service Directory
Service Broker
1. Initiate service request
2. UDDI find business service
3. Get URI for business service
4. Invoke business service with URI
5. Submit credentials for authentication
6. Submit credentials for authentication
7. Send SOAP messages via JAXM
8. Acknowledge SOAP messages via JAXM
9. Consolidate results
10. Return result to service requester using JAXM

FIG. 88



Typical Technology Used
Hyrbrid integration methods
Preagreed message formats/APIs
XML Web Services
HTTP/S GET or POST

When to Use
Brokering lowest price of similar services with a single front-end (Service-Provider neutral)
Loosely coupled process & technical integration
Large number of trading partners
Price-sensitive & homogeneous products

FIG. 89

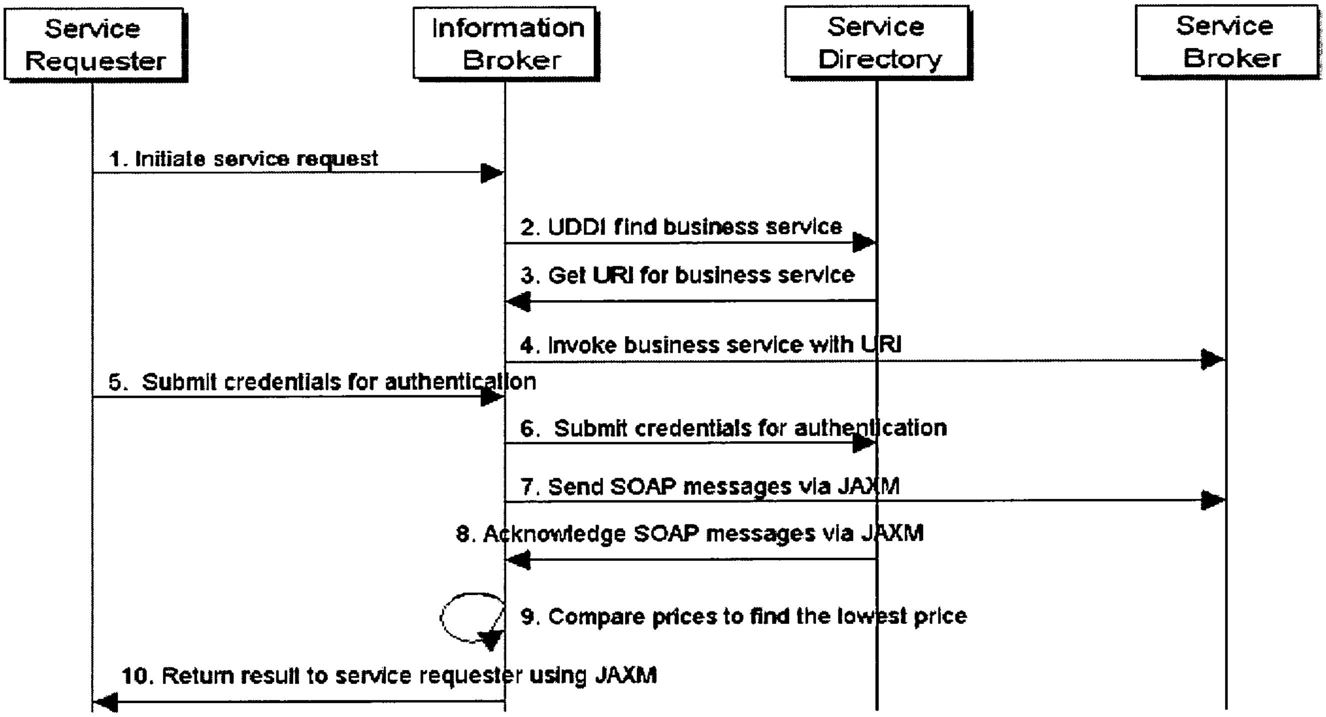

Service Requester
Information Broker
Service Directory
Service Broker
1. Initiate service request
2. UDDI find business service
3. Get URI for business service
4. Invoke business service with URI
5. Submit credentials for authentication
6. Submit credentials for authentication
7. Send SOAP messages via JAXM
8. Acknowledge SOAP messages via JAXM
9. Compare prices to find the lowest price
10. Return result to service requester using JAXM

FIG. 90

Integration Patterns	When to Use	Benefits	Consideration
Application to Application	Point-to-point exchange	Tight integration	Limited scalability
Standard Build	Strong branding Strong urge to standardize	Reduce deployment effort Standardized service, faster deployment with no customization	Consensus on standard builds
Hub-Spoke Replication Federated Replication Multi-step Application Integration	Hub-spoke business model Intra-enterprise integration	Flexible workflow integration Reliable and consistent multi-step application integration	Inter-enterprise integration with many customization options
Data Exchange	Large number of partners to integrate with heterogeneous platforms & standards	Accommodating differences in standards/interfaces	Emerging standards and technology
Closed Process Integration Open Process Integration	Shared business processes Workflow-oriented services	Richer support for process integration Cohesive and tightly integrated services	Complexity for partners to agree and implement
Service Consolidation–Broker Integration Reverse Auction–Broker Integration	Single front-end for multiple Service Providers	Added values and Service-Provider neutral	Handling service failure of partners

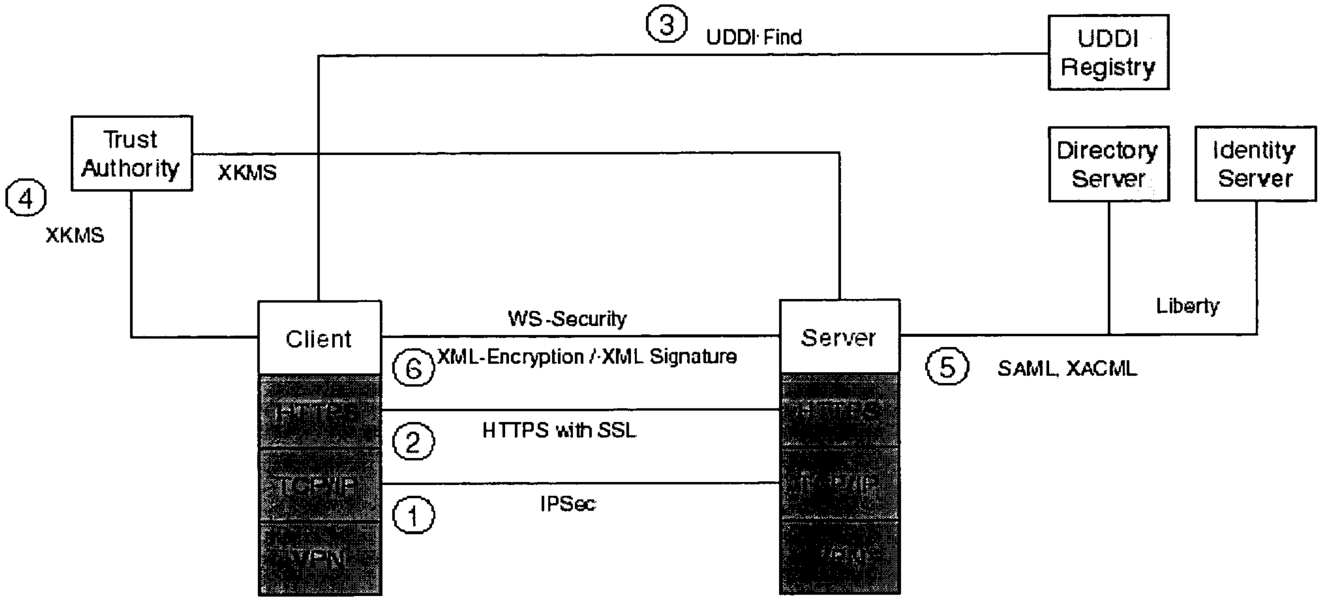
FIG. 91

Integration Patterns	Typical Technology Used	Typical Standards Used	Examples
Application to Application	Customized adapters EDI translator	Proprietary XML variants	Ariba Commerce One
Standard Build	Proprietary	Proprietary	Hexagon
Hub-Spoke Replication Federated Replication Multi-step Application Integration	EAI solutions, such as Amtrix, Mercator, and TIBCO	JMS, SOAP-JMS binding	eXonomy
Data Exchange	XML Web Services	XML and SOAP, UDDI, WSDL	AIG Visa Commerce
Closed Process Integration Open Process Integration	EAI solutions or middleware, such as Sun ONE Integration Server EAI edition, XML Web Services technology	BPEL4WS	
Service Consolidation–Broker Integration Reverse Auction–Broker Integration	Hybrid of any integration technology	Hybrid of any integration standards	Yahoo! Digilogistics (obsolete) CFOWeb Vcheq (obsolete) Bumiputra Commerce Bank

FIG. 92

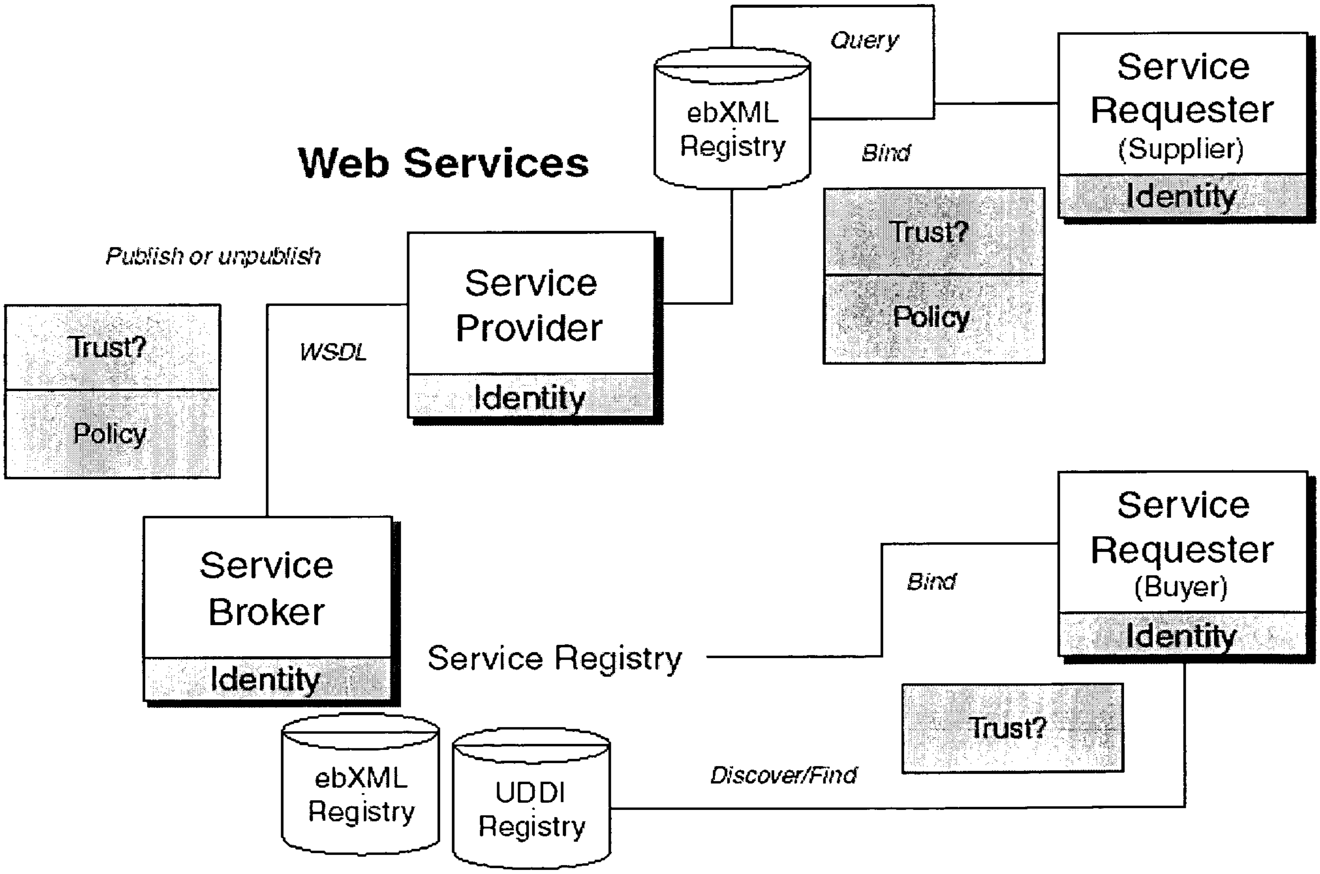
	Security Mechanism	Examples of Security Protection	Security Standards Specifications
Service Negotiation	Identity management Access control and policy management Single Sign-on	Liberty-compliant Identity Server Access control for XML messages Single Sign-on products	*Identity management*—Liberty 1.1, XML Key Management Specification (XKMS), WS-Federation *Entitlement*—SAML, XACML, WS-Authorization *Policy*—WS-Policy *Others*—WS-Secure Conversation, WS-Trust, WS-Privacy
Service Discovery	Service Registry security	UDDI Service Registry security features Protection for WSDL documents	UDDI WSDL
Transaction Routing	Messaging security	Data encryption Digital signature Key management and managing credentials	XML Encryption (XML-ENC) XML Signature (XML-DSIG) WS-Security XKMS
Transport	Data transport security	128-bit SSL with HTTPS Protocol security for FTP, SMTP, and so forth	HTTPS HTTPR IPSec
Internet	Network connectivity security	Leased line or router-level encryption Virtual Private Network (VPN) gateways	
Platform	Operating system security Penetration testing Key exchanges between hosts	Solaris OE™ hardening Linux Operating System (OS) hardening Windows OS hardening Professional Penetration Testing	

FIG. 93



3
UDDI Find
UDDI Registry
Trust Authority
XKMS
4
XKMS
Directory Server
Identity Server
Liberty
Client
WS-Security
Server
6
XML-Encryption / XML Signature
5
SAML, XACML
HTTPS
HTTPS
2
HTTPS with SSL
TCP/IP
TCP/IP
1
IPSec
VPN
VPN

FIG. 94



Query
ebXML Registry
Service Requester (Supplier)
Identity
Web Services
Bind
Trust?
Policy
Publish or unpublish
Service Provider
Identity
Trust?
Policy
WSDL
Service Requester (Buyer)
Identity
Bind
Service Broker
Identity
Service Registry
Trust?
Discover/Find
ebXML Registry
UDDI Registry

FIG. 95

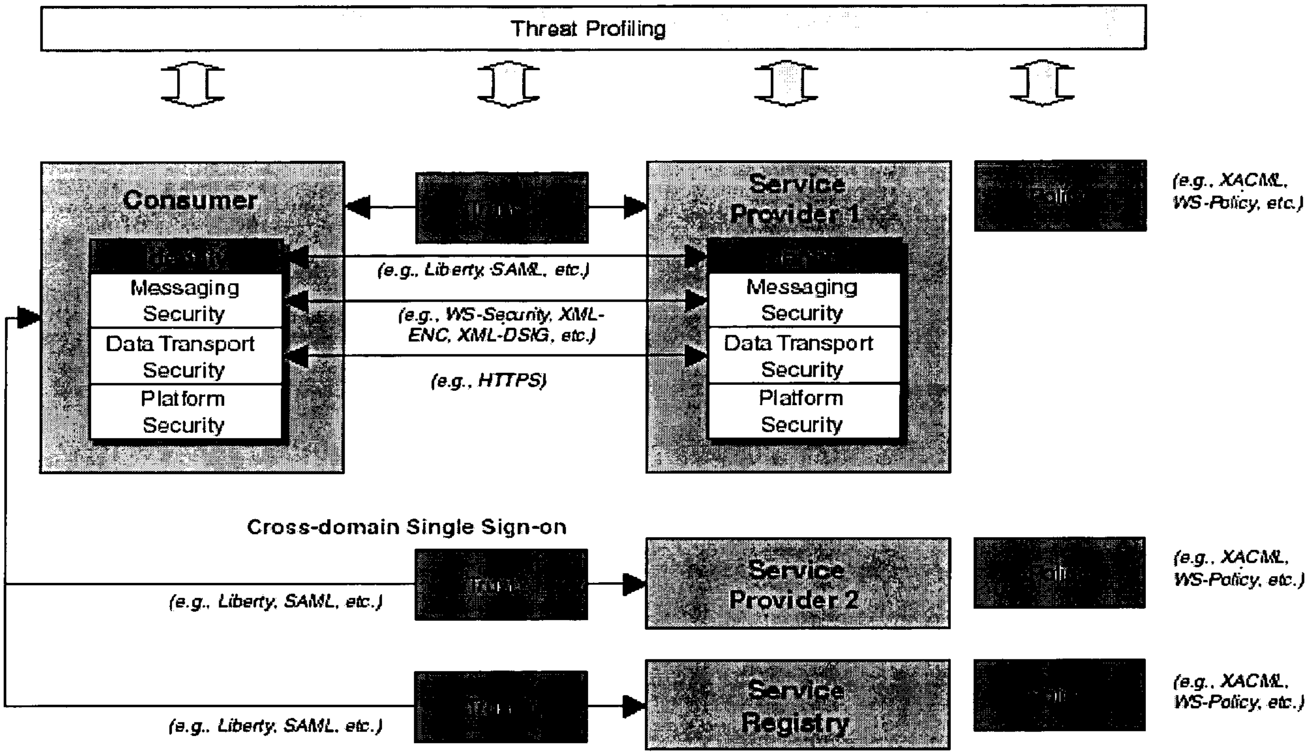
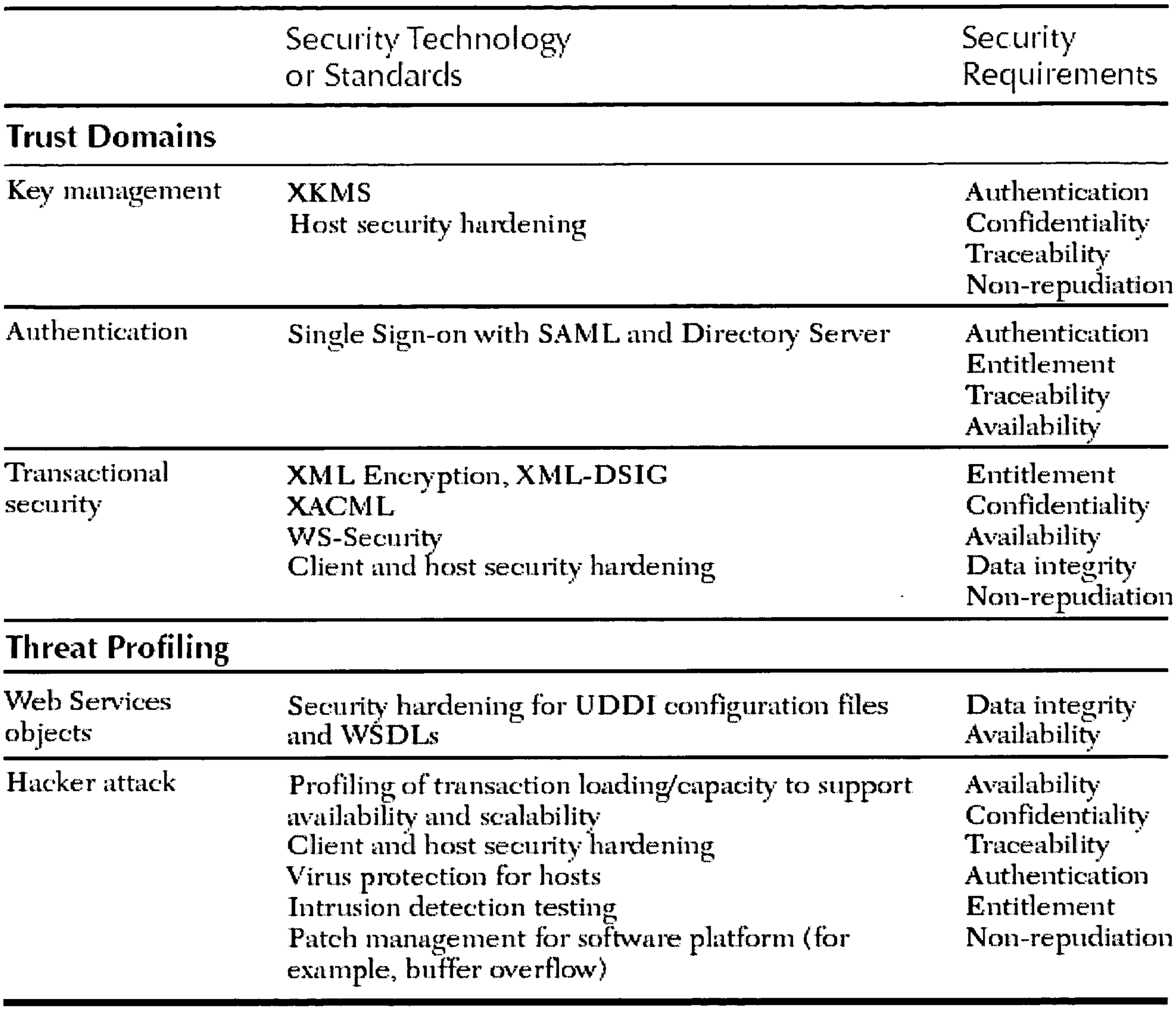
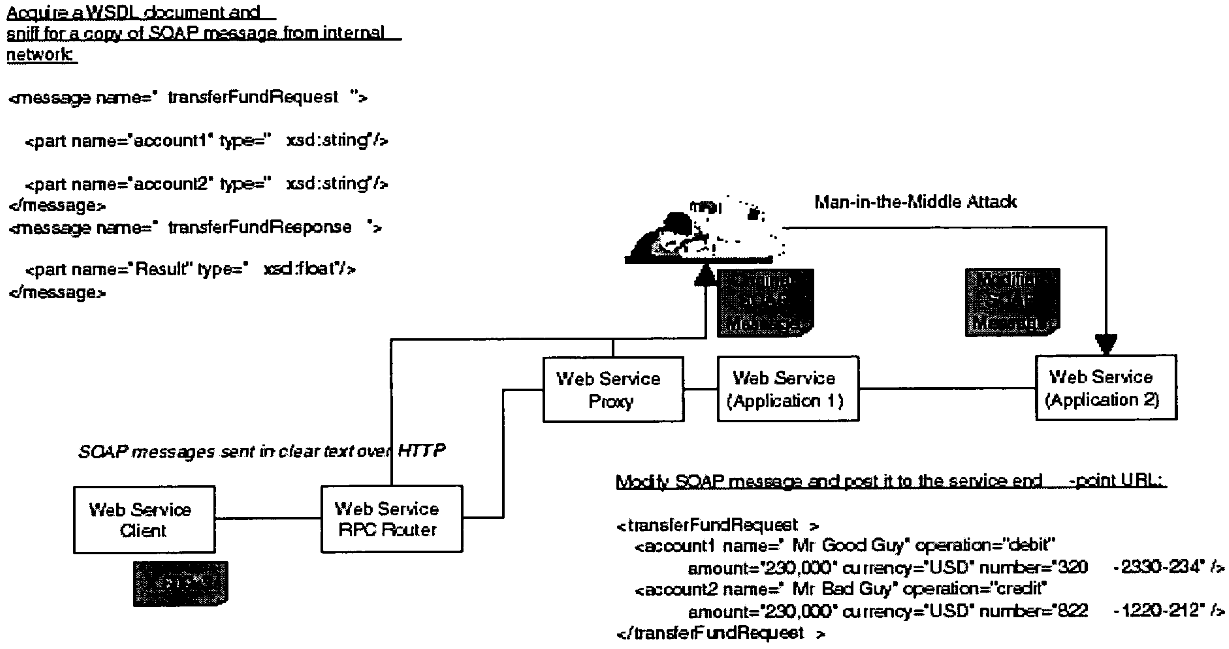

Threat Profiling
Consumer
Messaging Security
Data Transport Security
Platform Security
Service Provider 1
Messaging Security
Data Transport Security
Platform Security
(e.g., XACML, WS-Policy, etc.)
(e.g., Liberty SAML, etc.)
(e.g., WS-Security, XML-ENC, XML-DSIG, etc.)
(e.g., HTTPS)
Cross-domain Single Sign-on
Service Provider 2
(e.g., Liberty, SAML, etc.)
(e.g., XACML, WS-Policy, etc.)
Service Registry
(e.g., Liberty, SAML, etc.)
(e.g., XACML, WS-Policy, etc.)

FIG. 96



	Security Technology or Standards	Security Requirements
Trust Domains		
Key management	XKMS Host security hardening	Authentication Confidentiality Traceability Non-repudiation
Authentication	Single Sign-on with SAML and Directory Server	Authentication Entitlement Traceability Availability
Transactional security	XML Encryption, XML-DSIG XACML WS-Security Client and host security hardening	Entitlement Confidentiality Availability Data integrity Non-repudiation
Threat Profiling		
Web Services objects	Security hardening for UDDI configuration files and WSDLs	Data integrity Availability
Hacker attack	Profiling of transaction loading/capacity to support availability and scalability Client and host security hardening Virus protection for hosts Intrusion detection testing Patch management for software platform (for example, buffer overflow)	Availability Confidentiality Traceability Authentication Entitlement Non-repudiation

FIG. 97


Acquire a WSDL document and sniff for a copy of SOAP message from internal network

```
<message name="  transferFundRequest  ">

   <part name="account1" type="   xsd:string"/>

   <part name="account2" type="   xsd:string"/>
</message>
<message name="  transferFundResponse  ">

   <part name="Result" type="   xsd:float"/>
</message>
```

Modify SOAP message and post it to the service end   -point URL:

```
<transferFundRequest  >
   <account1 name="  Mr Good Guy" operation="debit"
         amount="230,000" currency="USD" number="320    -2330-234" />
   <account2 name="  Mr Bad Guy" operation="credit"
         amount="230,000" currency="USD" number="822    -1220-212" />
</transferFundRequest  >
```

FIG. 98

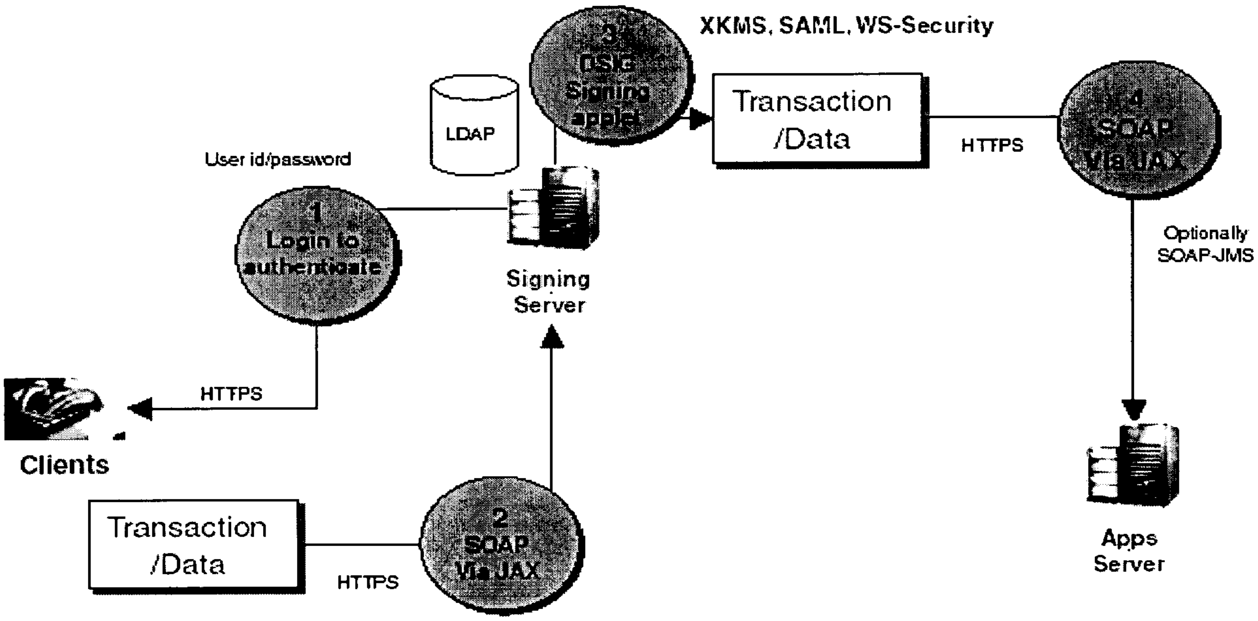


FIG. 99

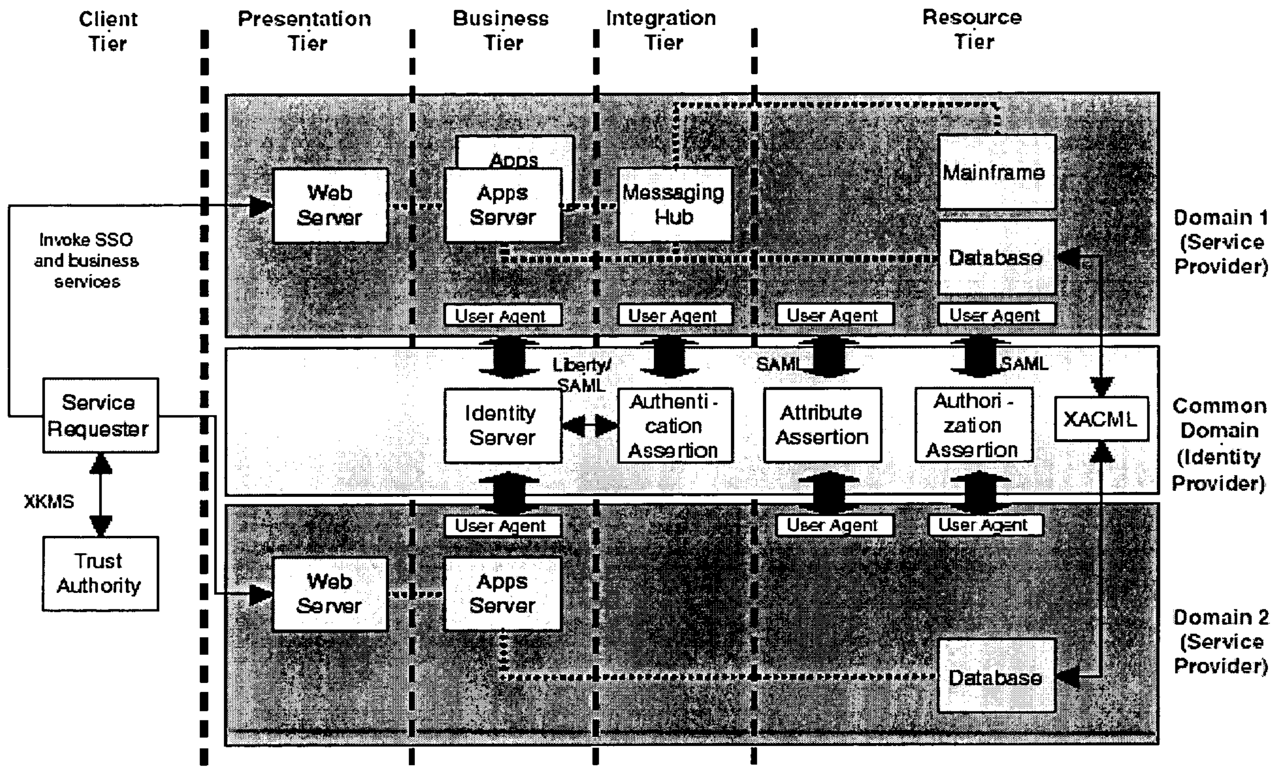


FIG. 100

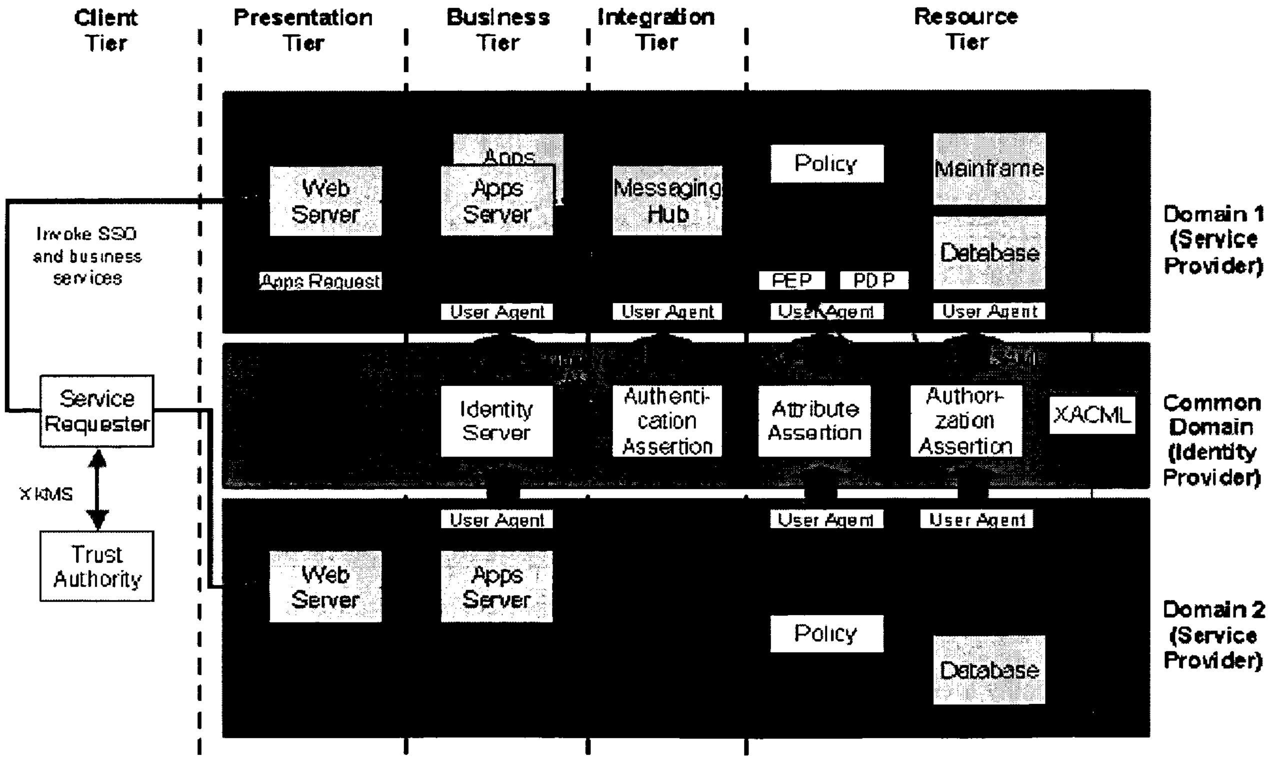


FIG. 101

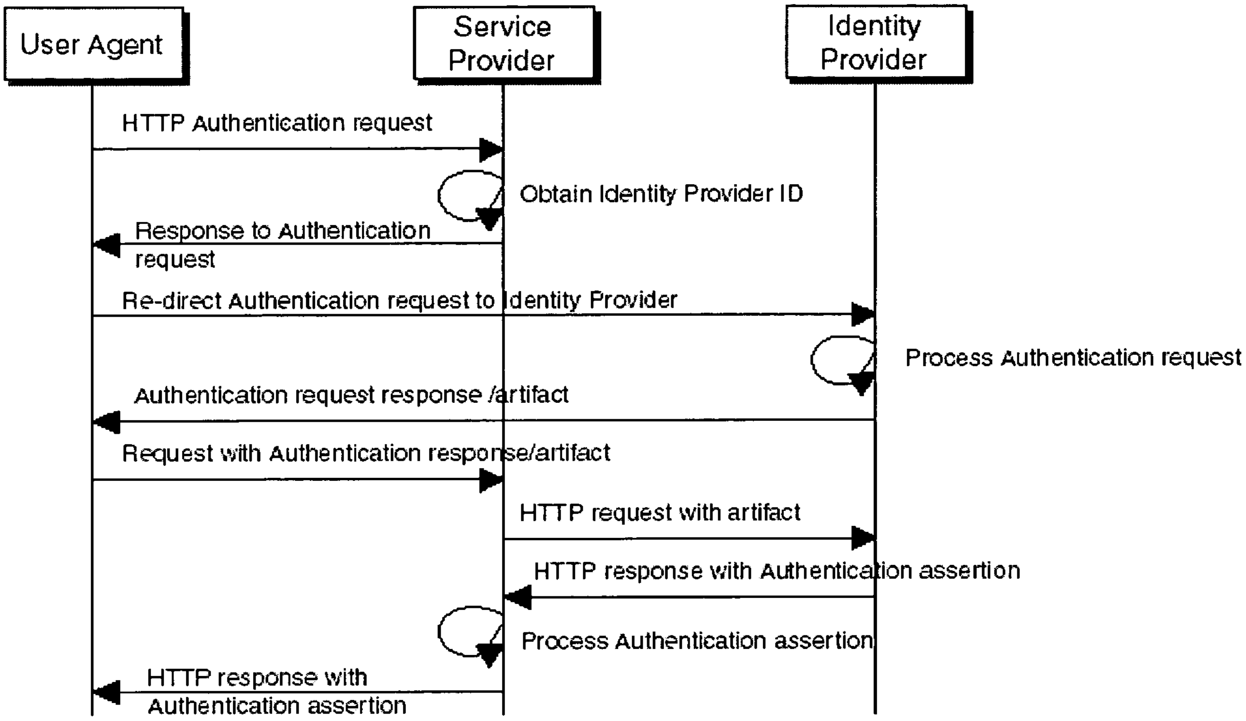

User Agent
Service Provider
Identity Provider
HTTP Authentication request
Obtain Identity Provider ID
Response to Authentication request
Re-direct Authentication request to Identity Provider
Process Authentication request
Authentication request response /artifact
Request with Authentication response/artifact
HTTP request with artifact
HTTP response with Authentication assertion
Process Authentication assertion
HTTP response with Authentication assertion

FIG. 102

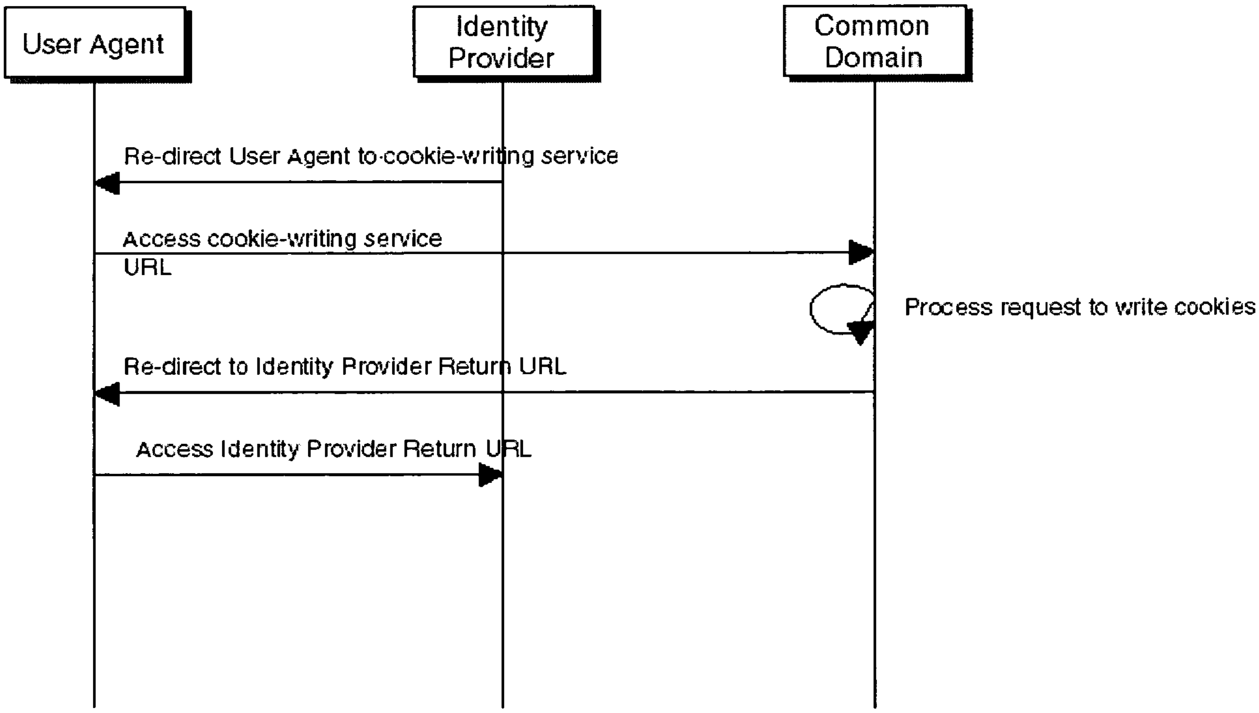

User Agent
Identity Provider
Common Domain
Re-direct User Agent to cookie-writing service
Access cookie-writing service URL
Process request to write cookies
Re-direct to Identity Provider Return URL
Access Identity Provider Return URL

FIG. 103

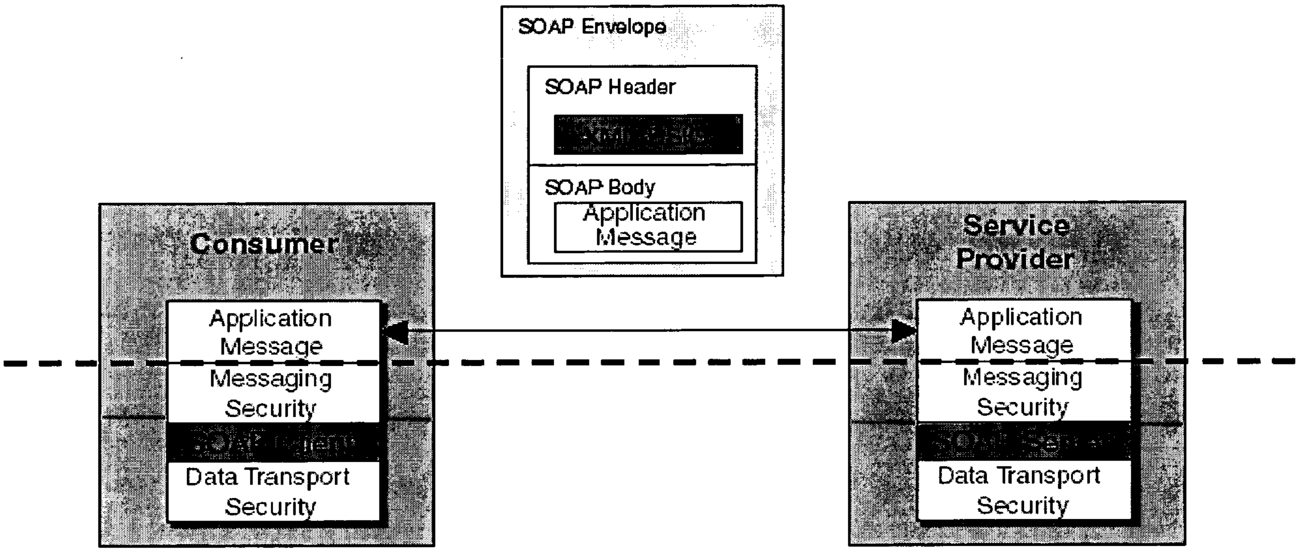
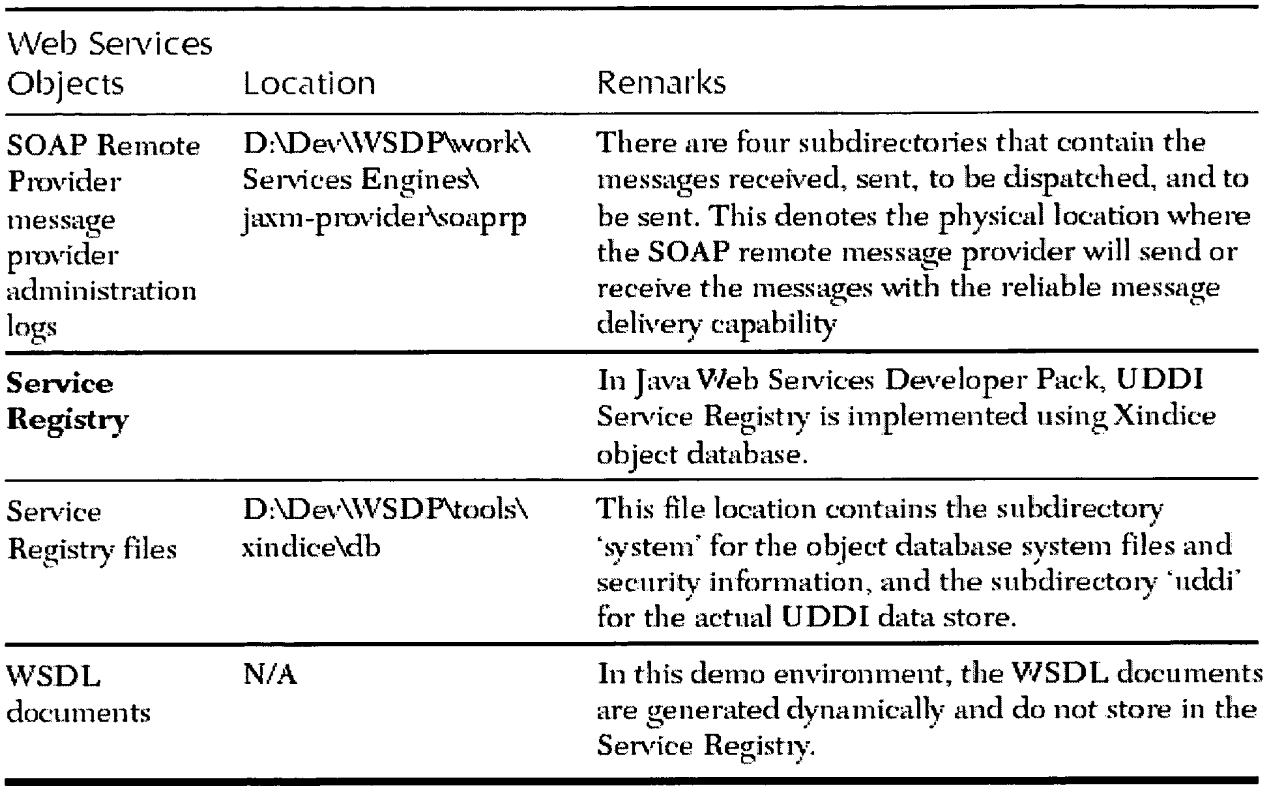

SOAP Envelope
SOAP Header
SOAP Body
Application Message
Consumer
Application Message
Messaging Security
SOAP Client
Data Transport Security
Service Provider
Application Message
Messaging Security
SOAP Server
Data Transport Security

FIG. 104

Web Services Objects	Location	Remarks
Web Container		In this example, this is Apache Tomcat 4.x.
User access control list	D:\Dev\WSDP\conf\ tomcat-users.xml	This file contains the user names, user passwords, and roles that are allowed to access and execute resources under the Web Container.
Server configuration file	D:\Dev\WSDP\conf\ server.xml	This file contains the server configuration (for example, port number) for running the Tomcat server.
Log Files		
Web Container log files	D:\Dev\WSDP\logs	In this example, Tomcat log files are used. This directory contains log files for Tomcat server (Catalina.out), server administration log (localhost_admin_log*.logand access_log*.log and services_log*.log), as well as Service Registry log (xindice.log).
Developer tool log files	D:\Dev\WSDP\logs\ jwsdp_log*.log	In this example, Java Web Services Developer Pack's log files are shown.
Service Registry update activity log file	D:\Dev\WSDP\tools\ xindice\logs\xindice.log	In this example, the Xindice database activity log file is used.
Message Provider		
ebXML message provider administration logs	D:\Dev\WSDP\work\ Services Engines\ jaxm-provider\ebxml	There are four subdirectories that contain the messages received, sent, to be dispatched, and to be sent. This denotes the physical location where the JAXM message provider will send or receive the messages with the reliable message delivery capability.

FIG. 105A



Web Services Objects	Location	Remarks
SOAP Remote Provider message provider administration logs	D:\Dev\WSDP\work\ Services Engines\ jaxm-provider\soaprp	There are four subdirectories that contain the messages received, sent, to be dispatched, and to be sent. This denotes the physical location where the SOAP remote message provider will send or receive the messages with the reliable message delivery capability
Service Registry		In Java Web Services Developer Pack, UDDI Service Registry is implemented using Xindice object database.
Service Registry files	D:\Dev\WSDP\tools\ xindice\db	This file location contains the subdirectory 'system' for the object database system files and security information, and the subdirectory 'uddi' for the actual UDDI data store.
WSDL documents	N/A	In this demo environment, the WSDL documents are generated dynamically and do not store in the Service Registry.

FIG. 105B

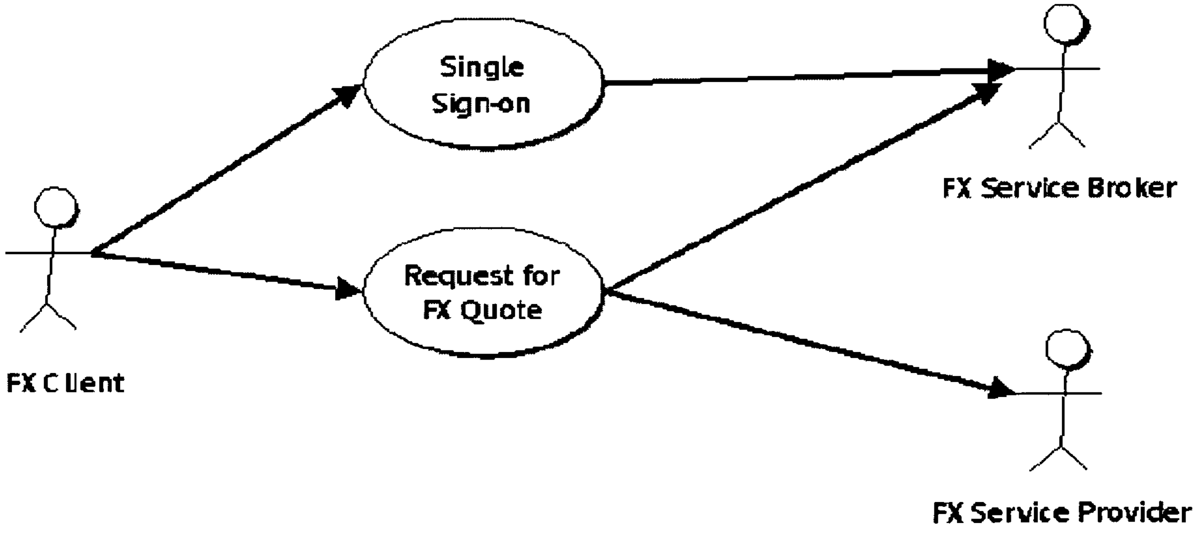


FIG. 106

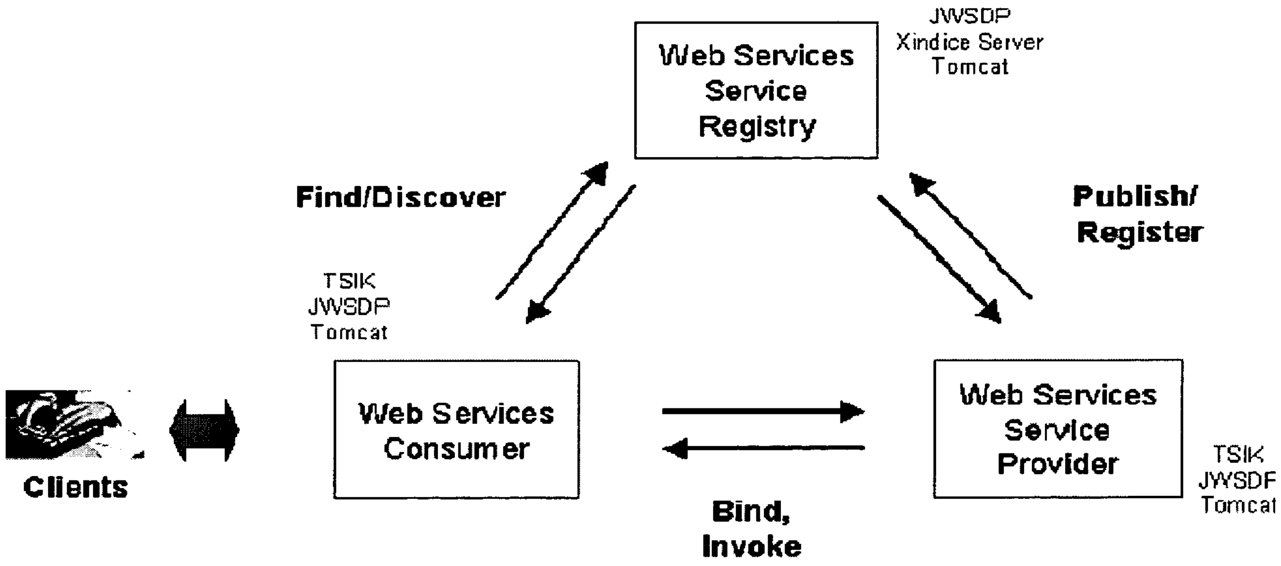


FIG. 107

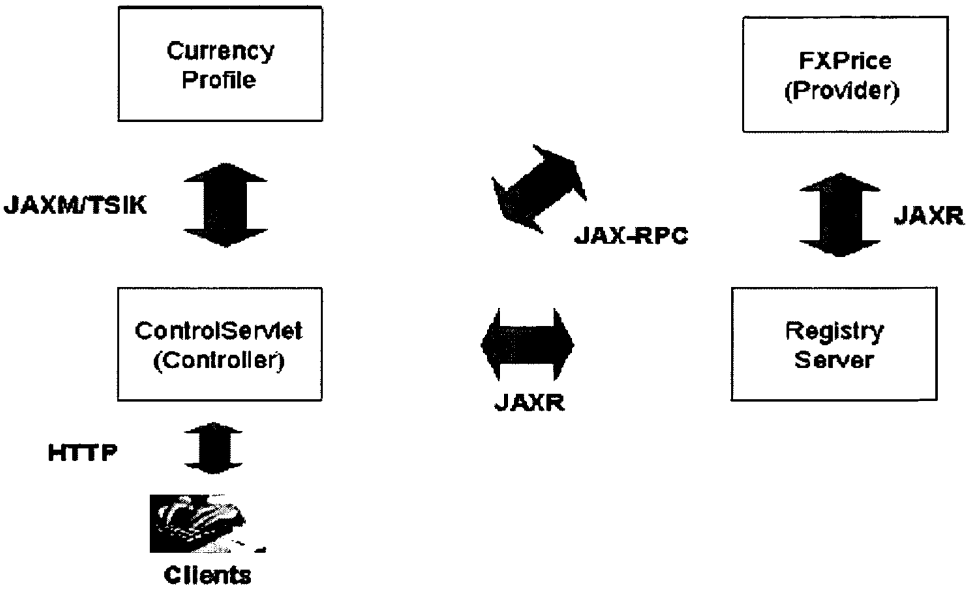


FIG. 108

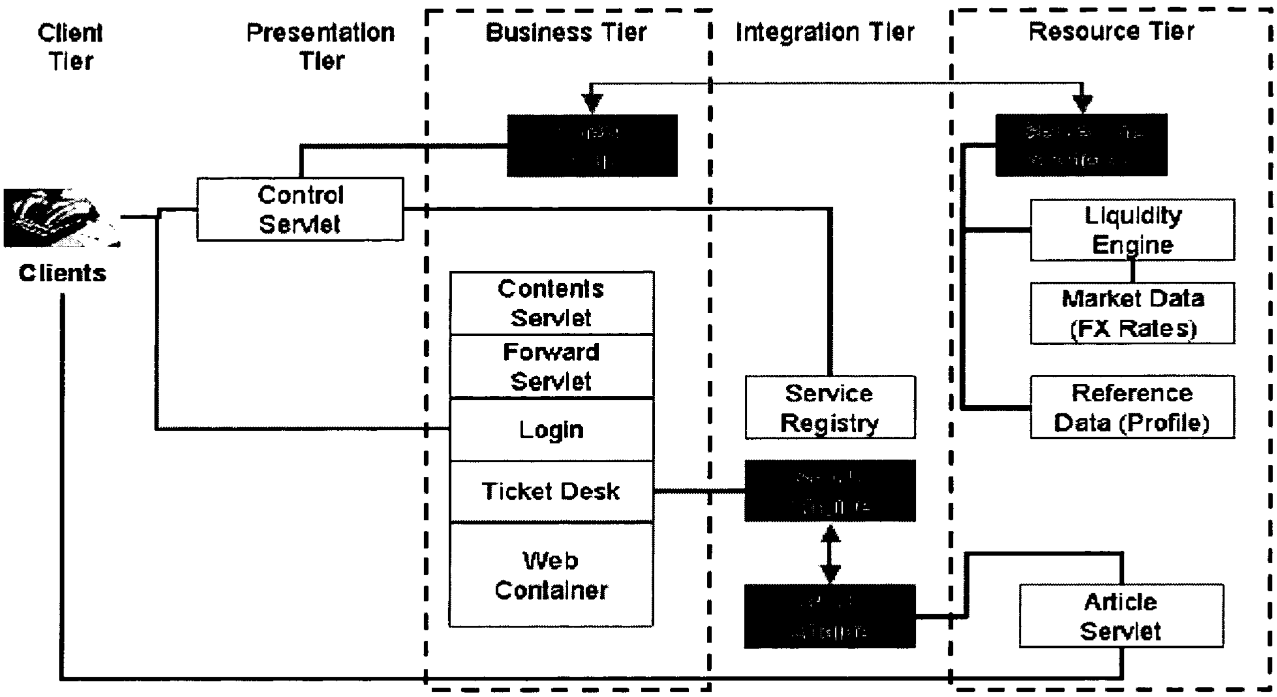

Client Tier
Presentation Tier
Business Tier
Integration Tier
Resource Tier
Clients
Control Servlet
Contents Servlet
Forward Servlet
Login
Ticket Desk
Web Container
Service Registry
Liquidity Engine
Market Data (FX Rates)
Reference Data (Profile)
Article Servlet

FIG. 109

			Tiers		
	Client	Presentation	Business	Integration	Resource
Application Platform Layer	User id and password are used for authentication.	Control Servlet uses HTML and JSP for presentation and inquiry. JSPs can be cached to enhance performance.	Java beans are used to implement some of the business logic. The remote FX Quote Service is a black box, accessible via JAX-RPC.	N/A	N/A
Virtual Platform Layer	HTTP HTTPS with SSL can be used for better security.	HTTP HTTPS with SSL can be used for better security.	JAXM-TSIK Message Provider provides secure messaging transport for SOAP messages over HTTP.	JAX-RPC, JAXM are used to integrate different remote services.	JAXR is used to access the Service Registry.
Upper Platform Layer	In the future, 128-bit SSL can be used for better security.	HTTP load balancing can be used for better scalability.	N/A	In the future, server clustering can be used for availability.	In the future, server clustering can be used for availability.
Lower Platform Layer	Basic Operating System security is provided with id and password.	N/A	N/A	N/A	N/A
Hardware Platform Layer	SSL accelerator can be added in the future for faster performance when using HTTPS.	Reliability and securability can be enhanced in the future with server hardening, firewall configuration, and hardware clustering.	Reliability and securability can be enhanced in the future with server hardening, firewall configuration, and hardware clustering.	N/A	Reliability and securability can be enhanced in the future with server hardening, firewall configuration, and hardware clustering.

FIG. 110

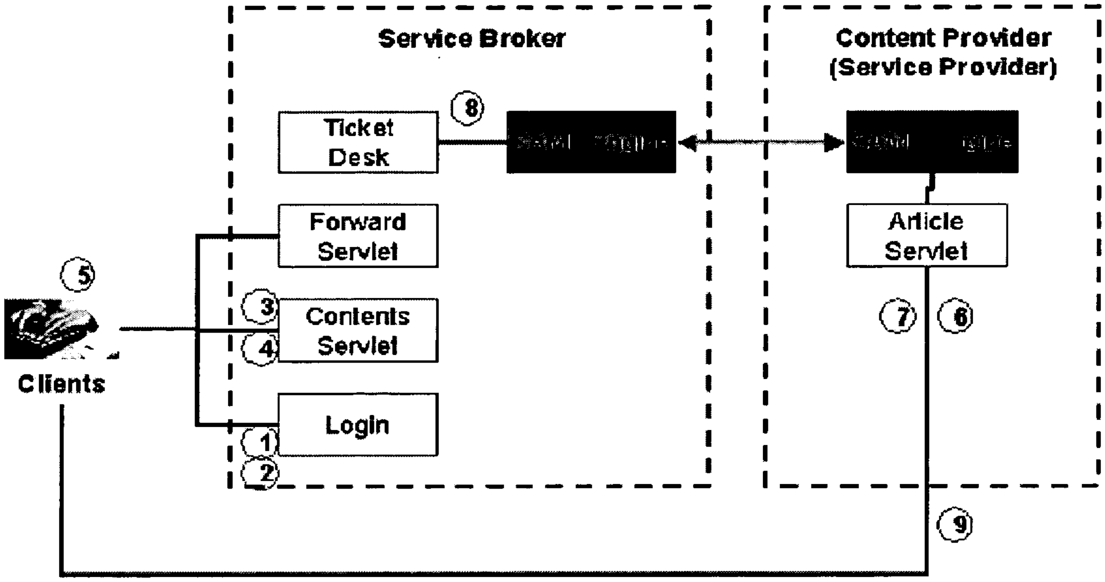


FIG. 111

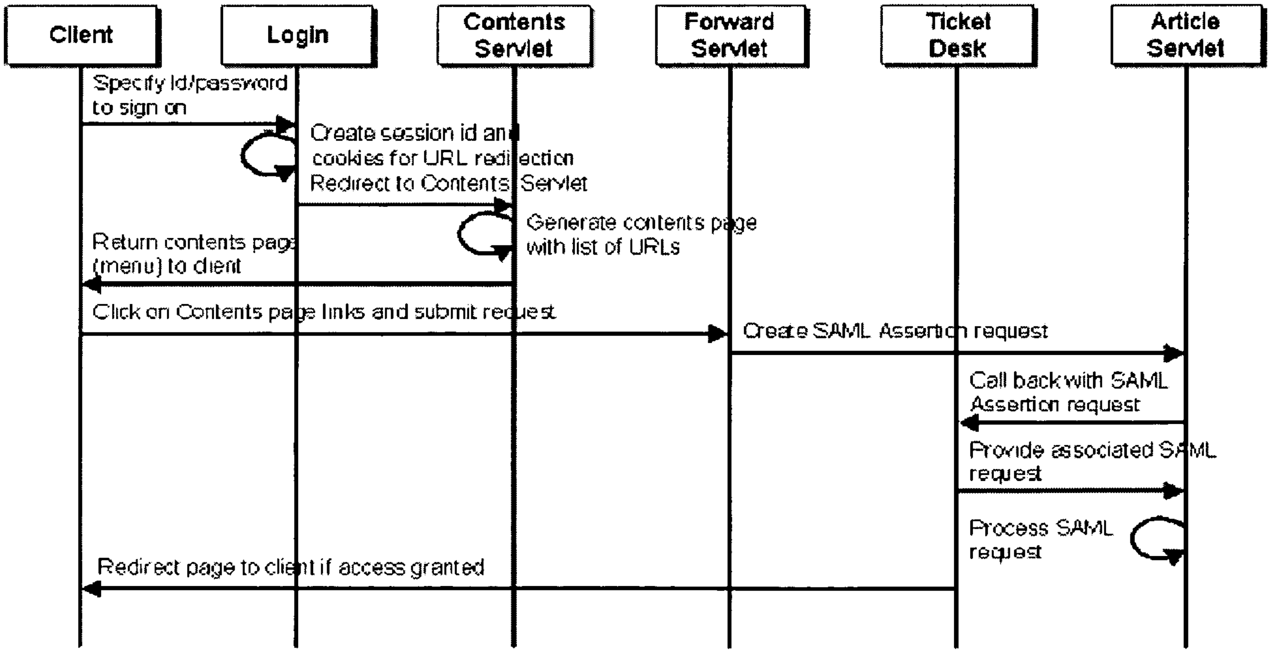

Client
Login
Contents Servlet
Forward Servlet
Ticket Desk
Article Servlet
Specify id/password to sign on
Create session id and cookies for URL redirection Redirect to Contents Servlet
Generate contents page with list of URLs
Return contents page (menu) to client
Click on Contents page links and submit request
Create SAML Assertion request
Call back with SAML Assertion request
Provide associated SAML request
Process SAML request
Redirect page to client if access granted

FIG. 112

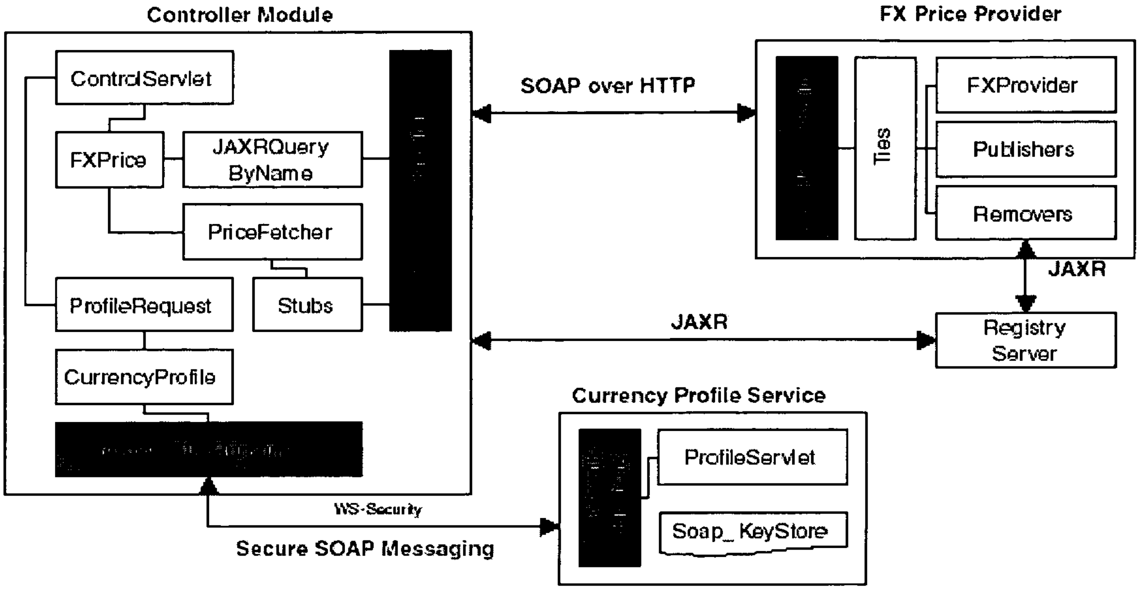


FIG. 113

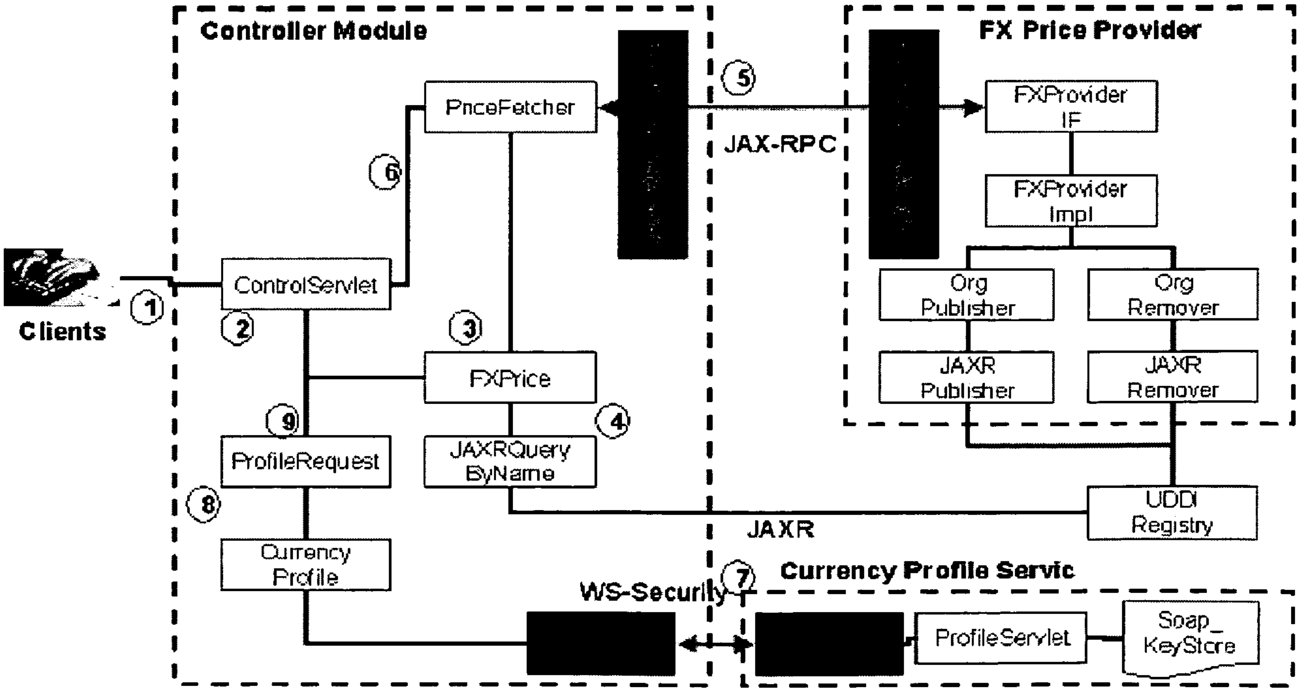


FIG. 114

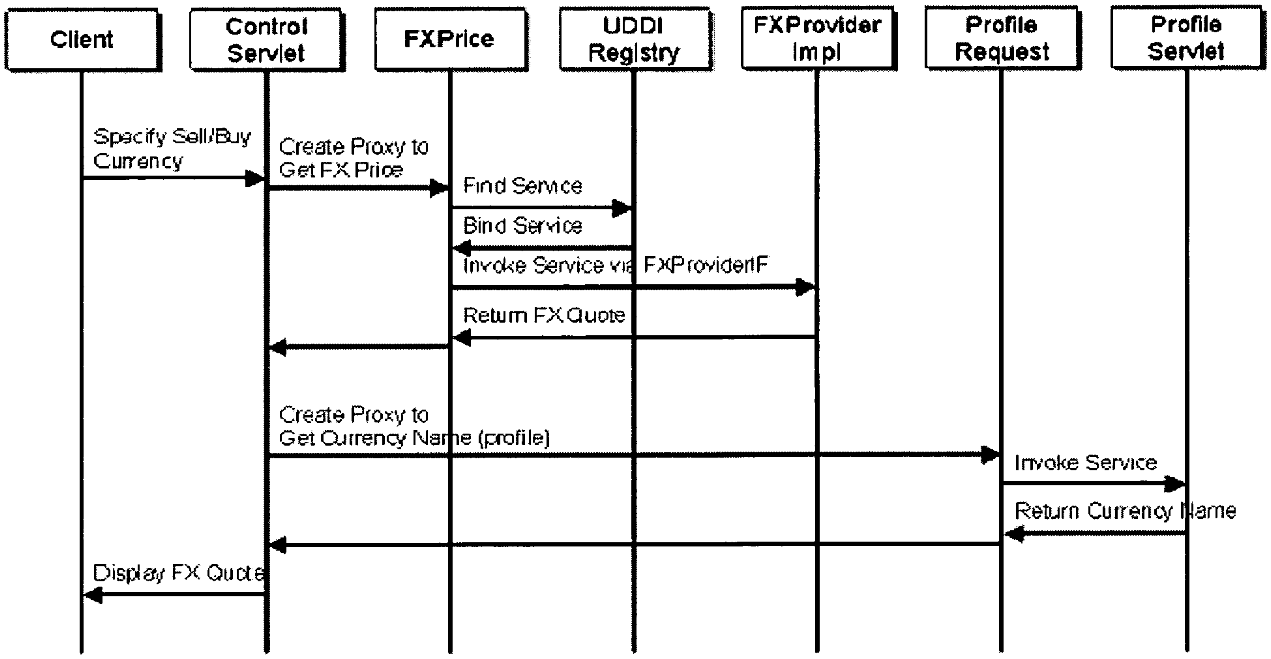


FIG. 115

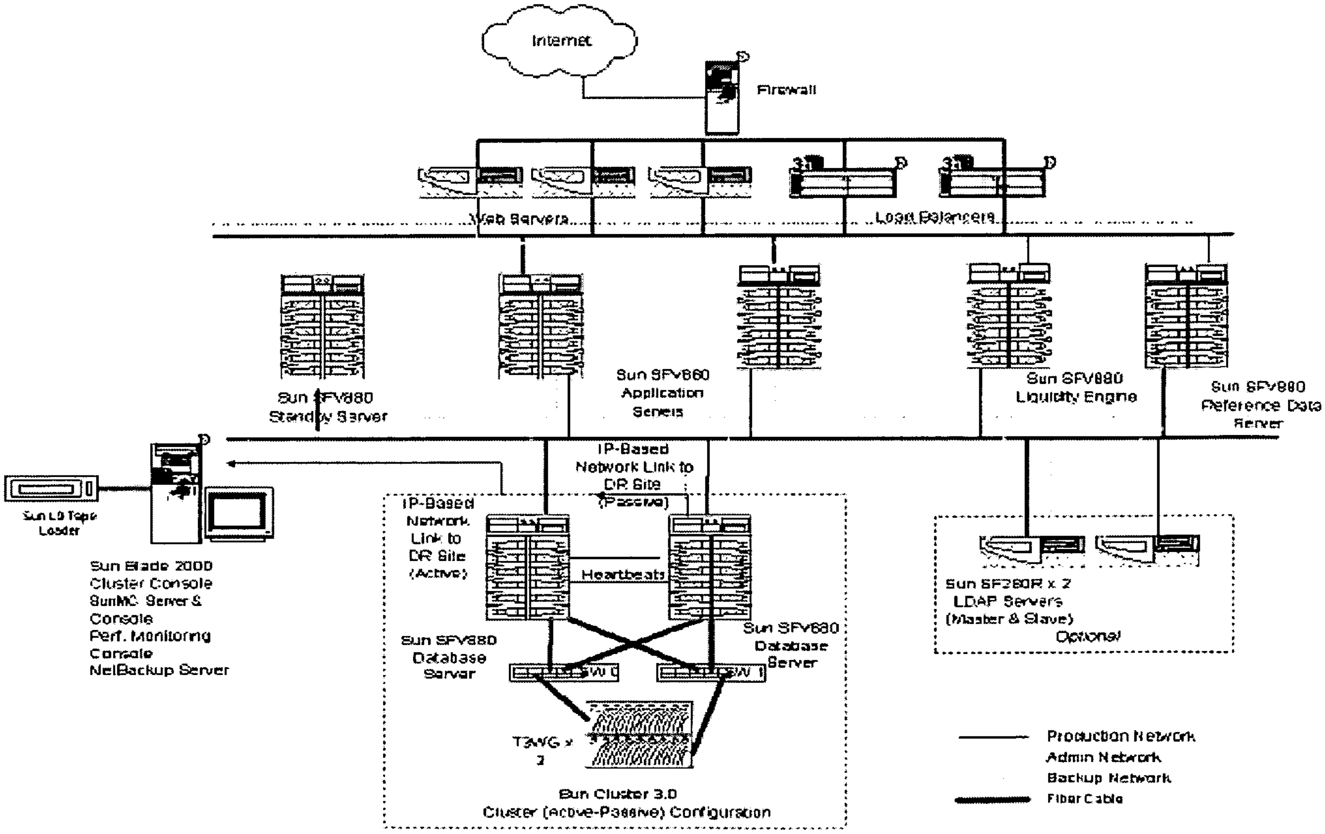

Internet
Firewall
Web Servers
Load Balancers
Sun SFV880 Standby Server
Sun SFV880 Application Servers
Sun SFV880 Liquidity Engine
Sun SFV880 Reference Data Server
Sun L9 Tape Loader
Sun Blade 2000 Cluster Console SunMC Server & Console Perf. Monitoring Console NetBackup Server
IP-Based Network Link to DR Site (Passive)
IP-Based Network Link to DR Site (Active)
Heartbeats
Sun SFV880 Database Server
Sun SFV880 Database Server
SW 0
SW 1
T3WG x 2
Sun Cluster 3.0 Cluster (Active-Passive) Configuration
Sun SF280R x 2 LDAP Servers (Master & Slave) Optional
Production Network
Admin Network
Backup Network
Fiber Cable

FIG. 116

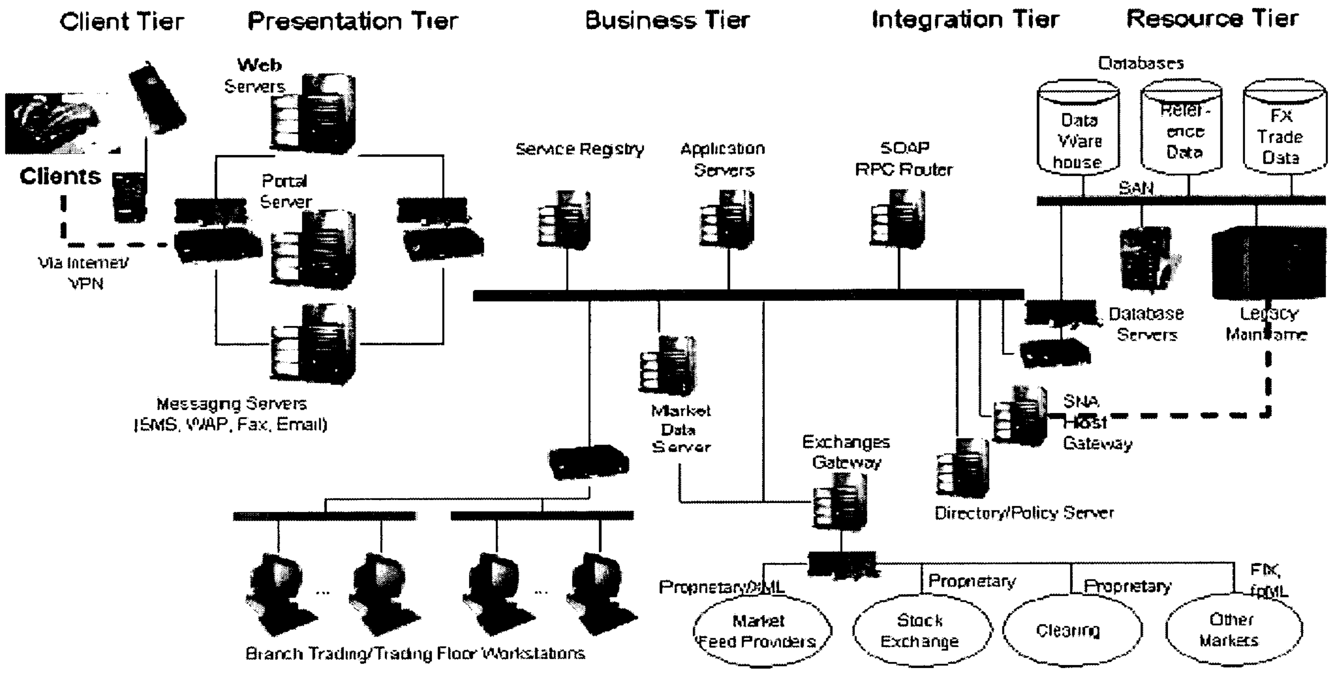

Client Tier
Presentation Tier
Business Tier
Integration Tier
Resource Tier
Web Servers
Clients
Via Internet/ VPN
Portal Server
Messaging Servers (SMS, WAP, Fax, Email)
Service Registry
Application Servers
SOAP RPC Router
Databases
Data Ware house
Refer-ence Data
FX Trade Data
SAN
Database Servers
Legacy Mainframe
SNA Host Gateway
Market Data Server
Exchanges Gateway
Directory/Policy Server
Proprietary/XML
Proprietary
Proprietary
FIX, fpML
Market Feed Providers
Stock Exchange
Clearing
Other Markets
...
...
Branch Trading/Trading Floor Workstations

FIG. 117

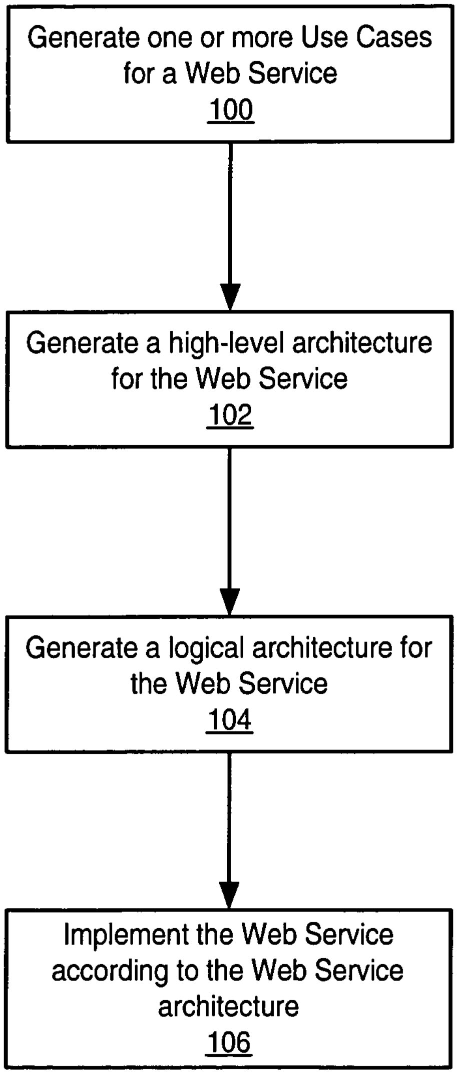

Generate one or more Use Cases for a Web Service
100
Generate a high-level architecture for the Web Service
102
Generate a logical architecture for the Web Service
104
Implement the Web Service according to the Web Service architecture
106

FIG. 118

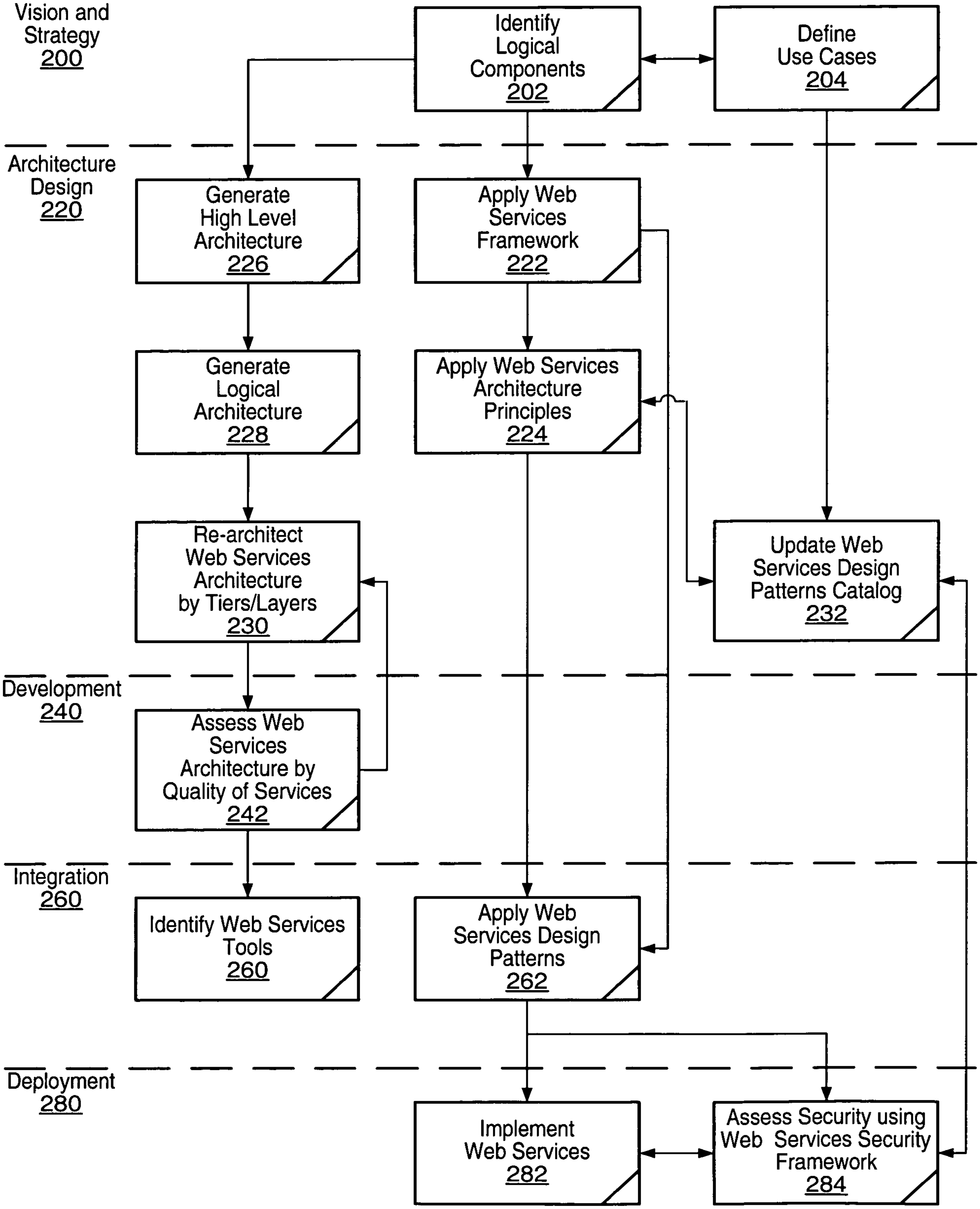

Vision and Strategy 200
Identify Logical Components 202
Define Use Cases 204
Architecture Design 220
Generate High Level Architecture 226
Apply Web Services Framework 222
Generate Logical Architecture 228
Apply Web Services Architecture Principles 224
Re-architect Web Services Architecture by Tiers/Layers 230
Update Web Services Design Patterns Catalog 232
Development 240
Assess Web Services Architecture by Quality of Services 242
Integration 260
Identify Web Services Tools 260
Apply Web Services Design Patterns 262
Deployment 280
Implement Web Services 282
Assess Security using Web Services Security Framework 284

FIG. 119

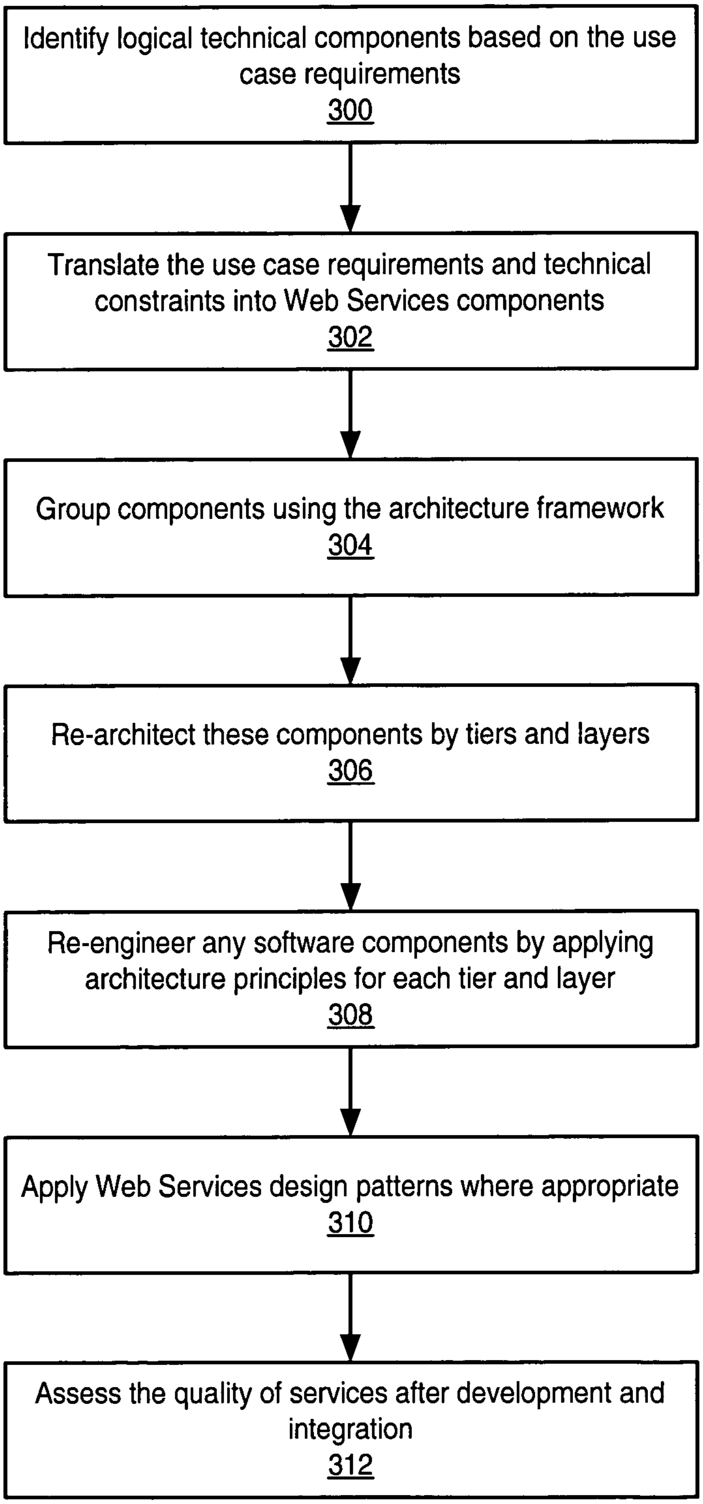

Identify logical technical components based on the use case requirements
300
Translate the use case requirements and technical constraints into Web Services components
302
Group components using the architecture framework
304
Re-architect these components by tiers and layers
306
Re-engineer any software components by applying architecture principles for each tier and layer
308
Apply Web Services design patterns where appropriate
310
Assess the quality of services after development and integration
312

FIG. 120

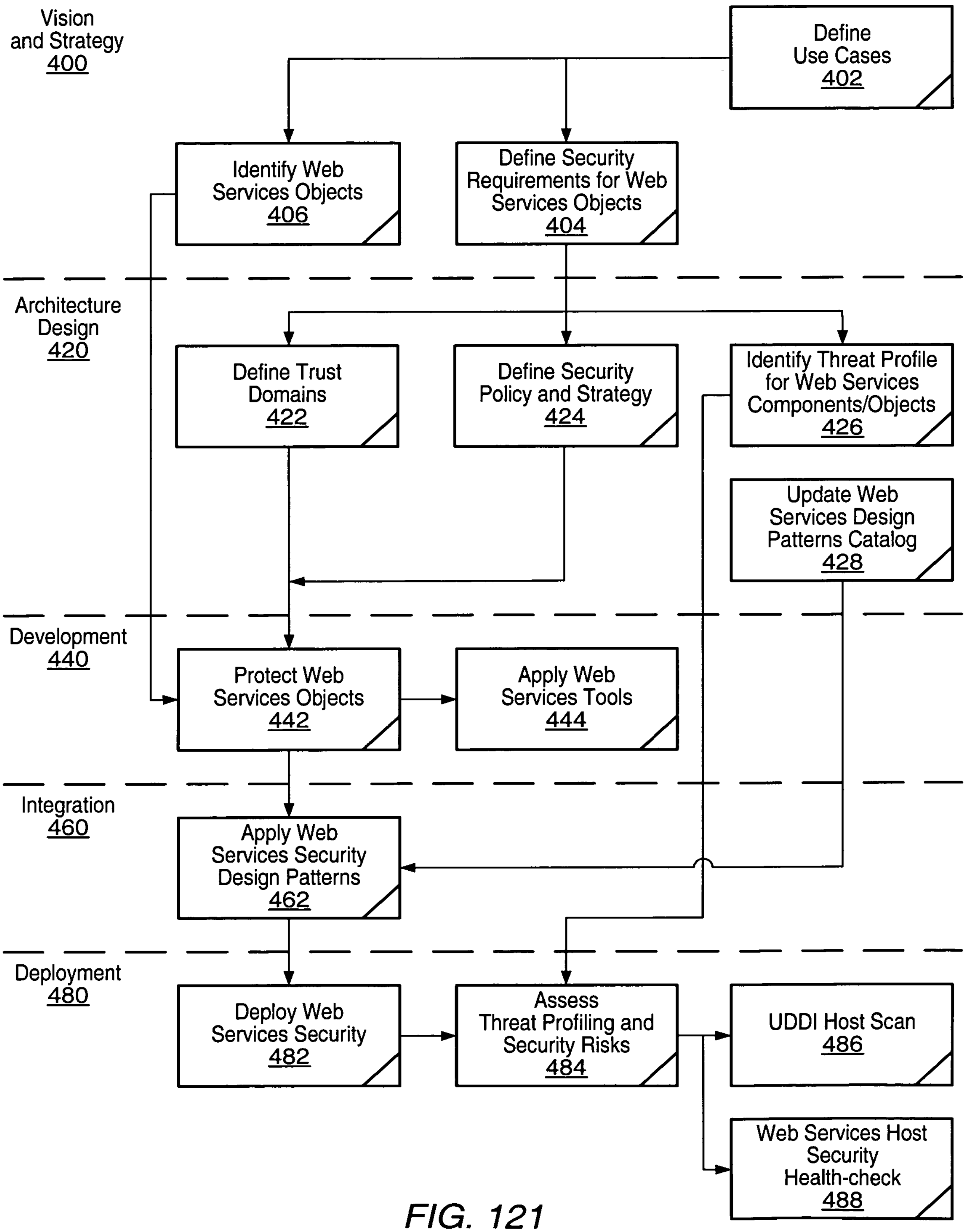

Vision
and Strategy
400
Define
Use Cases
402
Identify Web
Services Objects
406
Define Security
Requirements for Web
Services Objects
404
Architecture
Design
420
Define Trust
Domains
422
Define Security
Policy and Strategy
424
Identify Threat Profile
for Web Services
Components/Objects
426
Update Web
Services Design
Patterns Catalog
428
Development
440
Protect Web
Services Objects
442
Apply Web
Services Tools
444
Integration
460
Apply Web
Services Security
Design Patterns
462
Deployment
480
Deploy Web
Services Security
482
Assess
Threat Profiling and
Security Risks
484
UDDI Host Scan
486
Web Services Host
Security
Health-check
488

FIG. 121

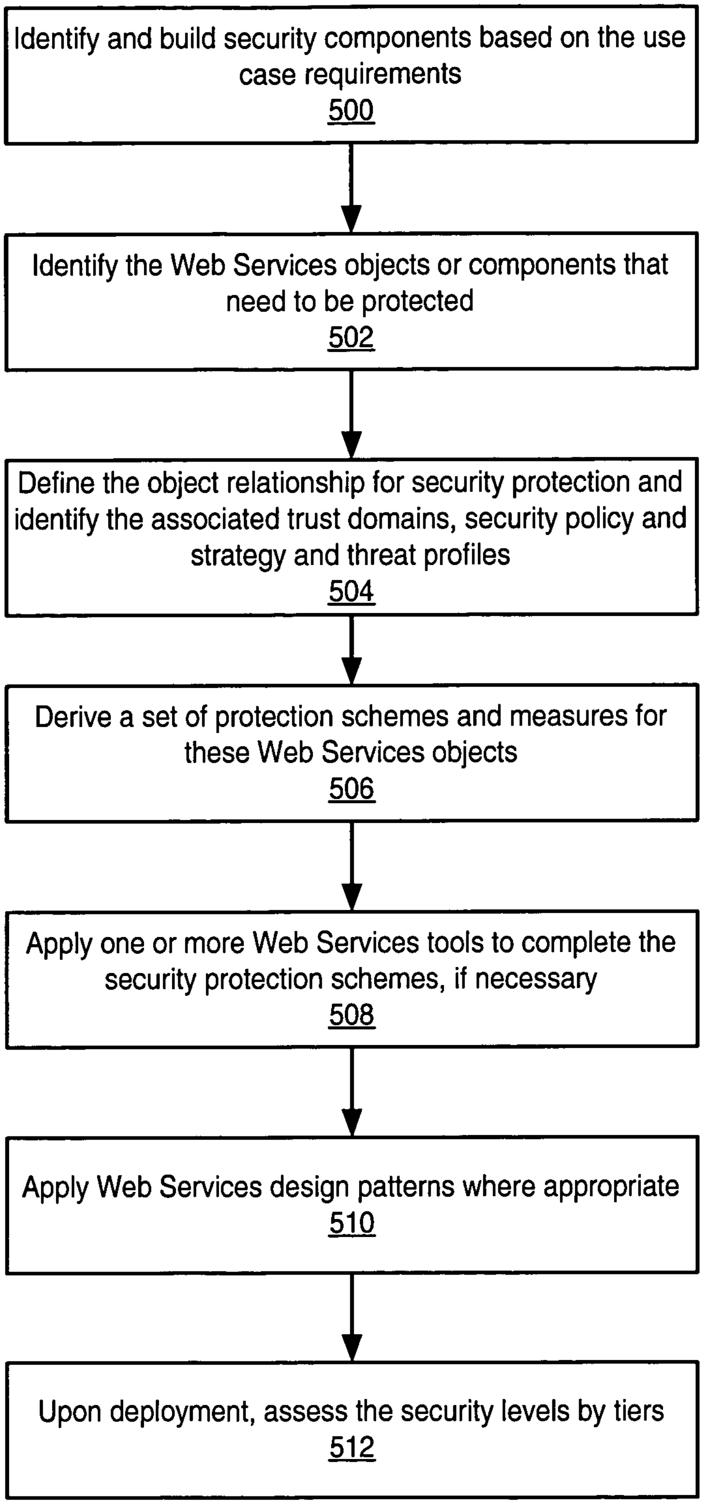

Identify and build security components based on the use case requirements
500
Identify the Web Services objects or components that need to be protected
502
Define the object relationship for security protection and identify the associated trust domains, security policy and strategy and threat profiles
504
Derive a set of protection schemes and measures for these Web Services objects
506
Apply one or more Web Services tools to complete the security protection schemes, if necessary
508
Apply Web Services design patterns where appropriate
510
Upon deployment, assess the security levels by tiers
512

FIG. 122

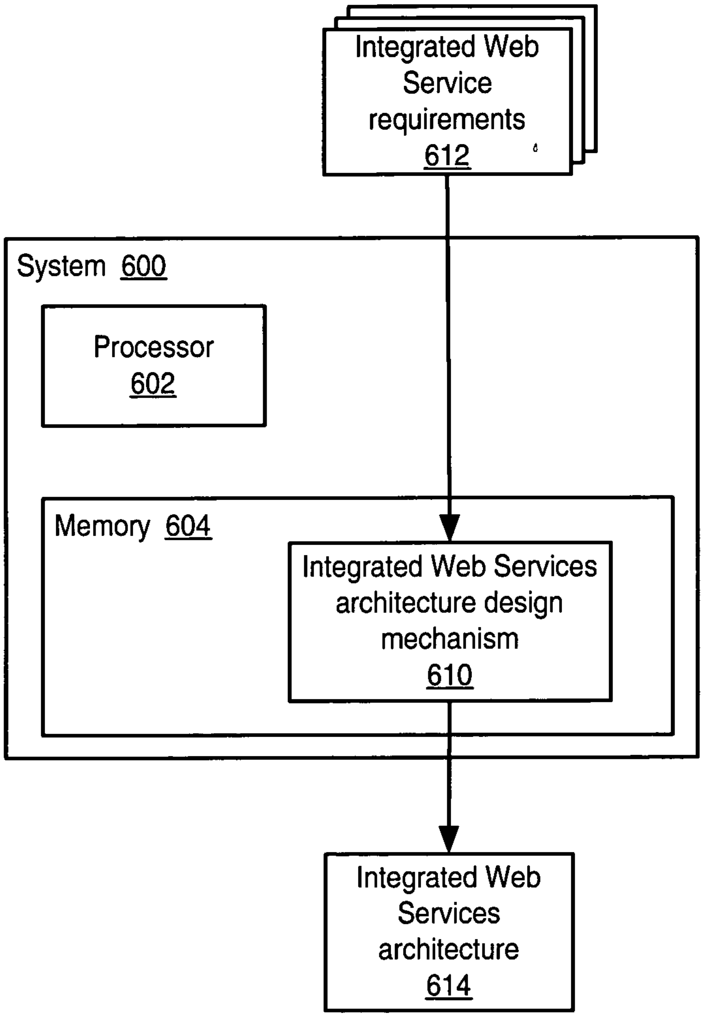

Integrated Web Service requirements 612
System 600
Processor 602
Memory 604
Integrated Web Services architecture design mechanism 610
Integrated Web Services architecture 614

FIG. 123

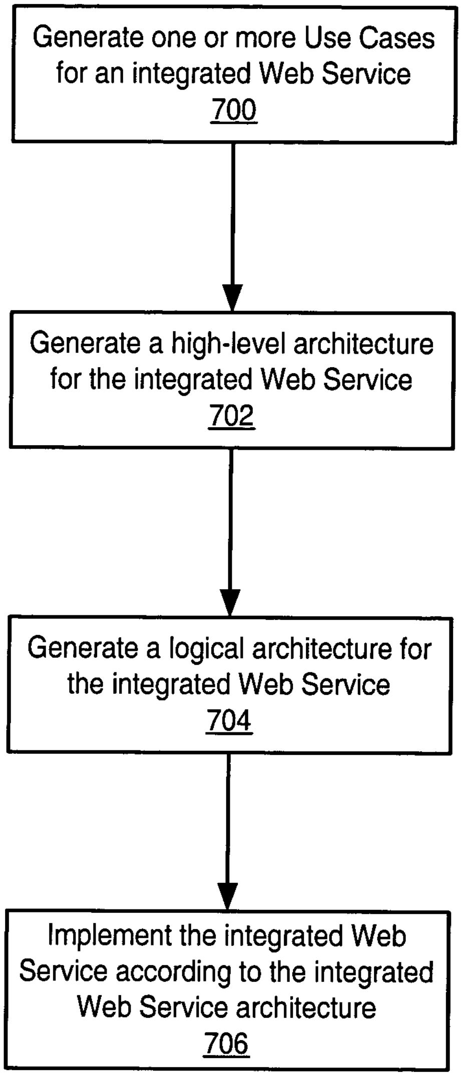

Generate one or more Use Cases for an integrated Web Service
700
Generate a high-level architecture for the integrated Web Service
702
Generate a logical architecture for the integrated Web Service
704
Implement the integrated Web Service according to the integrated Web Service architecture
706

FIG. 124

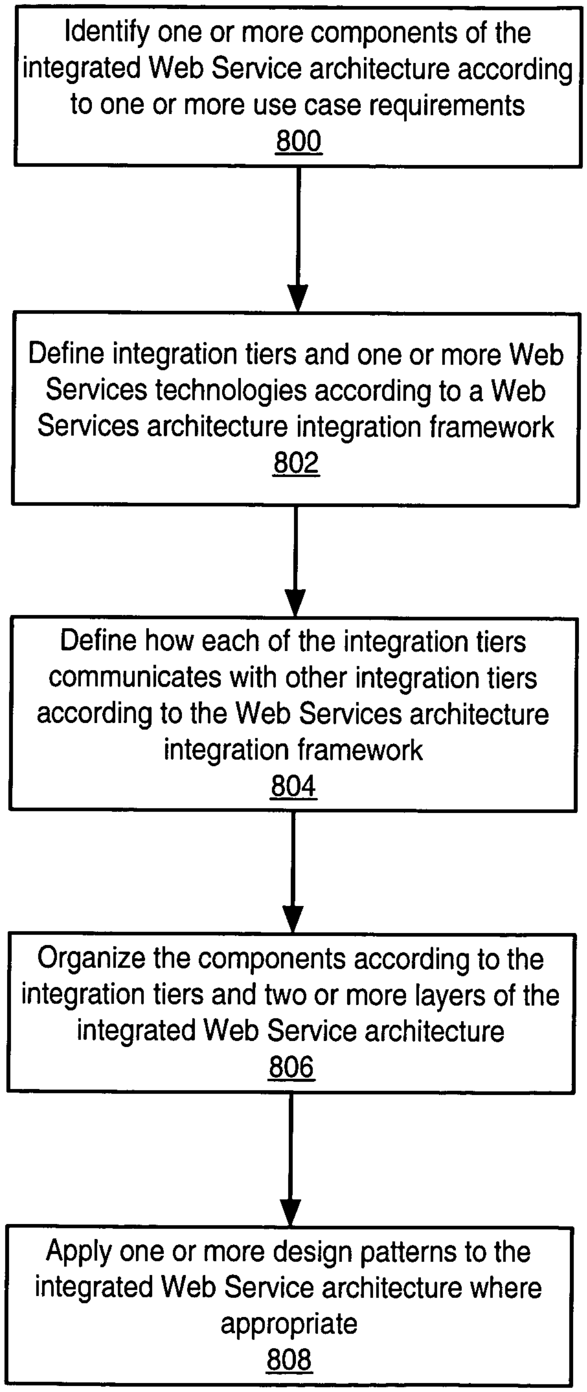

Identify one or more components of the integrated Web Service architecture according to one or more use case requirements
800
Define integration tiers and one or more Web Services technologies according to a Web Services architecture integration framework
802
Define how each of the integration tiers communicates with other integration tiers according to the Web Services architecture integration framework
804
Organize the components according to the integration tiers and two or more layers of the integrated Web Service architecture
806
Apply one or more design patterns to the integrated Web Service architecture where appropriate
808

FIG. 125